PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2001-075854**
(43)Date of publication of application : **23.03.2001**

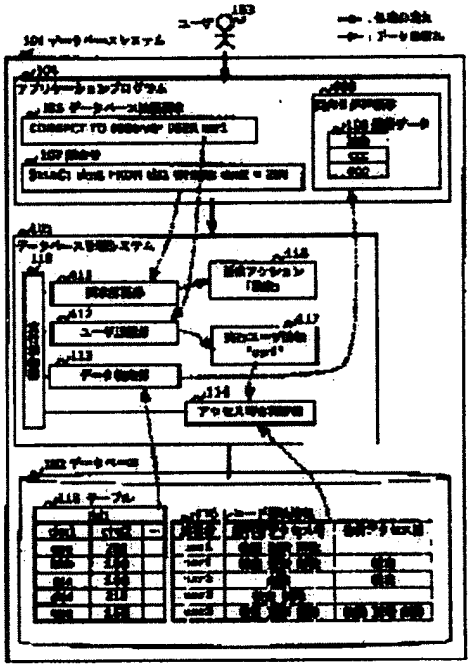
(51)Int.CI. G06F 12/00
G06F 12/14

(21)Application number : **11-246776**
(22)Date of filing : **31.08.1999**

(71)Applicant : **HITACHI LTD**
(72)Inventor : **KOBAYASHI TAKA**
TORII SHUNICHI
TSUCHIDA MASASHI

(54) METHOD AND SYSTEM FOR DATA MANAGEMENT, AND STORAGE MEDIUM WITH DATA MANAGEMENT PROGRAM STORED THEREIN

(57)Abstract:
PROBLEM TO BE SOLVED: To facilitate the access control for every user in a record unit in a data management system where a plurality of users share a database.
SOLUTION: This data management system 105 managing a database 102 has record attribute information being information on each record stored in the database other than the column value of the column of a record defined by a user. The record attribute information includes, for instance, a user identifier showing the owner of the record corresponding to the record attribute information, an access right showing what action is allowed to be performed about the record by a user being the owner of the record and an access right showing what action is allowed to be performed about the record by a user being different from the owner of the record. The record attribute information is referred to about an inquiry from an application program, the approval/denial of the access for the user to the record is judged, and consequently, the access control of a record unit for every user is performed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of

rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－75854
（P2001－75854A）
(43)公開日　平成13年３月23日(2001.3.23)

(51)Int.Cl.'	識別記号	FI		テーマコード'（参考）
G06F 12/00	537	G06F 12/00	537A	5B017
	520		520E	5B082
12/14	310	12/14	310K	

審査請求　未請求　請求項の数５　ＯＬ　（全 30 頁）

(21)出願番号　特願平11－246776

(22)出願日　平成11年８月31日(1999.8.31)

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　小林　挙
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所システム開発本部内

(72)発明者　鳥居　俊一
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所システム開発本部内

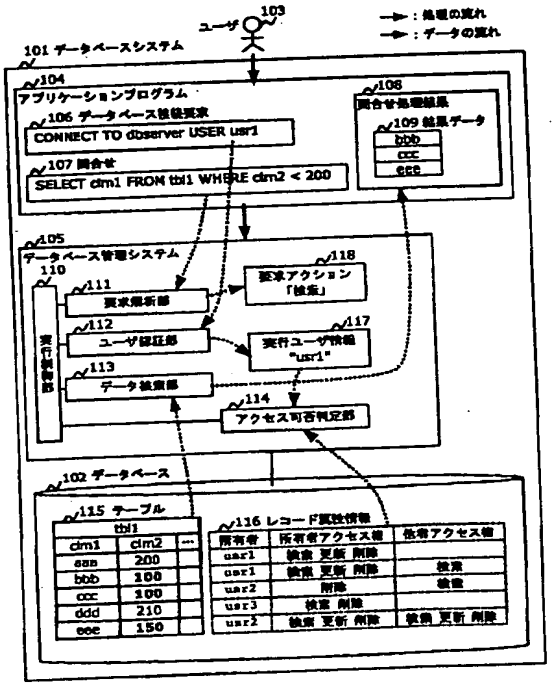
(74)代理人　100096954
弁理士　矢島　保夫

最終頁に続く

(54)【発明の名称】　データ管理方法、およびデータ管理システム、ならびにデータ管理プログラムを格納した記憶媒体

(57)【要約】
【課題】複数のユーザがデータベースを共用するデータ管理システムにおいて、レコード単位でユーザごとのアクセス制御を容易にすることを目的とする。
【解決手段】データベースを管理するデータベース管理システムが、ユーザが定義したレコードのカラムのカラム値以外に、データベースに保持した各レコードに関する情報であるレコード属性情報を持つようにする。レコード属性情報は、例えば、そのレコード属性情報に対応するレコードの所有者を示すユーザ識別子と、該レコードに対してレコードの所有者であるユーザがどのようなアクションを行うことを許可するかを示すアクセス権と、該レコードに対してレコードの所有者とは異なるユーザがどのようなアクションを行うことを許可するかを示すアクセス権とを含む。アプリケーションプログラムからの問合せに対しては、レコード属性情報を参照し、そのユーザの当該レコードへのアクセス可否を判定し、これによりユーザごとのレコード単位のアクセス制御を行う。

データ検索処理の概念図（図１）



1

【特許請求の範囲】
【請求項1】データへのアクセスを管理するデータ管理方法において、
データに対応付けられたアクセス制御に関する値を保持するデータ属性を管理し、
データ処理要求を入力すると、該データ処理要求がデータにアクセス可能であるかを該データに対応付けられた前記データ属性に基づいて判定し、
該判定の結果アクセス可能である場合に当該データに対して前記データ処理要求に基づいたデータ処理を行うことを特徴とするデータ管理方法。
【請求項2】請求項1に記載のデータ管理方法において、
前記データ属性の値が未指定の場合、あらかじめ指示された値を前記データ属性の値とみなすことを特徴とするデータ管理方法。
【請求項3】請求項1に記載のデータ管理方法において、
データ処理要求を行うユーザを識別するユーザ識別子と、データに対して該ユーザがアクセス可能であるか否かを示すアクセス権を、該データに対応付けがなされている前記データ属性に含むことを特徴とするデータ管理方法。
【請求項4】データへのアクセスを管理するデータ管理システムにおいて、
データ処理要求を行ったユーザを認識するユーザ認識手段と、
該データ処理要求を解析し、要求されているアクションを認識する要求解析手段と、
前記データ属性に、データに対してユーザがアクションを行えるか否かを示すアクセス権を保持して、該データ属性に基づいてデータへのアクセス可否を判定するアクセス可否判定手段を備えることを特徴とするデータ管理システム。
【請求項5】データへのアクセスを管理するデータ管理システムにおけるデータ管理プログラムを格納した記憶媒体であって、
前記データ管理プログラムは、
データに対応付けられたアクセス制御に関する値を保持するデータ属性を管理し、
データ処理要求を入力すると、該データ処理要求がデータにアクセス可能であるかを該データに対応付けられた前記データ属性に基づいて判定し、
該判定の結果アクセス可能である場合に当該データに対して前記データ処理要求に基づいたデータ処理を行うものであることを特徴とする記憶媒体。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】本発明は、データ管理方法、データ管理システム、および該システムに係るプログラムやデータを記憶した記憶媒体に関し、特に、レコード単位でユーザごとのアクセス制御を行うデータ管理の方法などに関する。

2

【0002】
【従来の技術】データベース用問合せ言語の標準規格であるSQLでは、ユーザごとのアクセス制御として、GRANT文により、テーブル単位で参照、更新、および削除などのアクセスの制御を行うことが規定されている。
【0003】これに対し、データベースシステムにおいて、レコード単位でユーザごとにアクセス制御を行う方法として、データベースに格納するレコードを構成するカラム値にユーザを識別する値を設定し、データベースを利用するプログラムが、そのユーザを識別する値によってアクセス制御する方法がある。そのような例としては、例えば、特開平10-124491号「文書共有整理システム、共有文書管理装置および文書アクセス装置」に開示された方法がある。
【0004】また、レコード単位でユーザごとにアクセス制御する方法としては、特開平6-348575号「データベース制御装置」に開示された方法がある。この方法では、カラム値の条件とユーザごとのアクセス権との対応を設定したアクセス権管理テーブルを用い、アクセス権管理テーブルの設定に従って、カラム値がある条件を満たした場合に、それに対応するユーザごとのアクセスを制御することにより、レコード単位の複雑なアクセス権制御を実現できるようにしている。
【0005】
【発明が解決しようとする課題】前記の従来技術では、データベースを管理するデータベース管理システムにおいて、レコードを構成する、ユーザが定義したカラムのカラム値以外の情報をもとにしてデータベース処理を制御することについて配慮されておらず、そのため、アクセス制御の判断のもとになる値をすべてカラム値で表現しレコードに含めてデータベースに格納し、データベースを利用するプログラムにおいてカラム値をもとにアクセス制御を行わなければならず、データベースを利用するプログラムの負荷が大きいという問題があった。
【0006】例えば、複数のユーザが1つのスキーマの1つのテーブルにレコードを登録し、あるユーザがそのテーブルから自分の作成したレコードのみを参照できるようにアクセス制御するには、データベースを利用するプログラムであるアプリケーションプログラムで、以下のようなアクセス制御を行う必要がある。
【0007】まず、テーブルの定義において、各レコードを作成したユーザを示すユーザ識別子を保持するカラムを用意しなければならない。そして、レコード登録において、アプリケーションプログラムは、ユーザ識別子を生成し、ユーザ識別子を保持するよう定義したカラムにそのユーザ識別子を格納するようにする。
【0008】そして、検索においては、次の2つの方法

によりアクセス制御を行う。

【0009】1つの方法は、データベースへの問合せ（データベース管理システムへの検索要求）で、前記のユーザ識別子を保持するカラムに検索を実行するユーザのユーザ識別子が保持されているレコードのみを検索するように問合せの条件を設定し、データベース管理システムでの一般的な条件による絞り込みにより、レコードを選択する方法である。

【0010】もう1つの方法は、データベースを検索して得られるレコードのうち、前記のユーザ識別子を保持するカラムに検索を実行したユーザのユーザ識別子が保持されているレコードのみをアプリケーションプログラムが選択する方法である。

【0011】このように、従来技術によれば、ユーザごとのアクセス制御は、すべてアプリケーションプログラムが実装しなければならない。

【0012】なお、同じユーザを示すためにユーザ識別子を統一したり、複数のテーブルで同様にユーザ識別を行うために整合性を保つことも、アプリケーションプログラムで保証しなければならない。

【0013】また、データベースに登録されるユーザ識別子は、データベース管理システムの管理外でアプリケーションプログラムが独自に作成した値であり、データベース管理システムは、それがユーザを識別するための値であることを認識しない。ユーザ識別子を保持するカラムも、データベース管理システムでは、ユーザを識別するための特別なカラムとして他のカラムと区別して取り扱うことはないので、データベース管理システムは、ユーザ識別子を保持したカラムをもとにユーザを認識することはない。従って、アクセス制御を行うアプリケーションプログラムを介さずに直接データベース管理システムを通してデータベースを参照するユーザ、あるいはユーザ識別子を故意に変更するユーザには、アクセス制御することができない。

【0014】また、SQLで規定されたユーザ管理をサポートするデータベース管理システムにおけるユーザ管理とは独立にアプリケーションプログラムでアクセス制御を行うことになるため、データベース管理システムで行われるアクセス制御とアプリケーションで行うアクセス制御とを、整合するように調整しなければならない。

【0015】また、1つのスキーマ内では同じ名称のテーブルを定義することができず、1つのテーブルを複数のユーザが共用した場合、テーブルにUNIQUE指定すると、異なるユーザが同一のレコードを登録できない。よって、同一の表を複数ユーザが共用して、ユーザごとに個人別にレコードを参照するようなことはできない。

【0016】従来技術では、上記の問題を解決する方法やシステムが開示されていない。

【0017】本発明の目的は、複数のユーザがデータベースを共用するデータ管理システムにおいて、レコード単位でユーザごとのアクセス制御を容易にすることにある。

【0018】

【課題を解決するための手段】上記目的を達成するため、本発明は、データへのアクセスを管理するデータ管理方法において、データに対応付けられたアクセス制御に関する値を保持するデータ属性を管理し、データ処理要求を入力すると、該データ処理要求がデータにアクセス可能であるかを該データに対応付けられた前記データ属性に基づいて判定し、該判定の結果アクセス可能である場合に当該データに対して前記データ処理要求に基づいたデータ処理を行うことを特徴とする。

【0019】また、前記データ属性の値が未指定の場合、あらかじめ指示された値を前記データ属性の値とみなすことを特徴とする。さらに本発明は、データ処理要求を行うユーザを識別するユーザ識別子と、データに対して該ユーザがアクセス可能であるか否かを示すアクセス権を、該データに対応付けがなされている前記データ属性に含むことを特徴とする。

【0020】また本発明は、データへのアクセスを管理するデータ管理システムにおいて、データ処理要求を行ったユーザを認識するユーザ認識手段と、該データ処理要求を解析し、要求されているアクションを認識する要求解析手段と、前記データ属性に、データに対してユーザがアクションを行えるか否かを示すアクセス権を保持して、該データ属性に基づいてデータへのアクセス可否を判定するアクセス可否判定手段を備えることを特徴とする。

【0021】さらに本発明は、データへのアクセスを管理するデータ管理システムにおけるデータ管理プログラムを格納した記憶媒体であって、前記データ管理プログラムは、データに対応付けられたアクセス制御に関する値を保持するデータ属性を管理し、データ処理要求を入力すると、該データ処理要求がデータにアクセス可能であるかを該データに対応付けられた前記データ属性に基づいて判定し、該判定の結果アクセス可能である場合に当該データに対して前記データ処理要求に基づいたデータ処理を行うものであることを特徴とする。

【0022】

【発明の実施の形態】以下、本発明の実施の形態を図面を用いて詳細に説明する。まず、本発明の実施形態のデータベースシステムを示し、本発明の原理を説明する。

【0023】図1は、本発明の実施形態のデータベースシステムの概要を示す概念図である。このデータベースシステム101は、データベース102へのユーザ103からのアクセスを管理するシステムである。

【0024】図1に示すように、本実施形態のデータベースシステム101は、アプリケーションプログラム（AP）104と、データベース管理システム105からなる。

【００２５】ＡＰ１０４は、データベース接続要求１０６をデータベース管理システム１０５に送信してデータベース管理システム１０５との接続を確立し、ユーザ１０３からの要求を表わす問合せ１０７をデータベース管理システム１０５に送信し、その要求に見合うデータベース処理の結果である問合せ処理結果１０８を受け取るプログラムである。

【００２６】データベース管理システム１０５は、データベース１０２を管理し、ＡＰ１０４からの問合せに見合うデータベース処理を行い、その処理結果である結果データ１０９を含む問合せ処理結果１０８をＡＰ１０４に返却するプログラムである。

【００２７】データベース管理システム１０５は、実行制御部１１０と、要求解析部１１１と、ユーザ認証部１１２と、データ検索部１１３と、アクセス可否判定部１１４からなる。

【００２８】実行制御部１１０は、データベース管理システム１０５内で行う処理の実行制御を行う。すなわち、実行制御部１１０は、データベース管理システム１０５で行う一連の処理の部分処理を、要求解析部１１１と、ユーザ認証部１１２と、データ検索部１１３と、アクセス可否判定部１１４で処理するように制御する。要求解析部１１１は、ＡＰ１０４からの要求を解析する。ユーザ認証部１１２は、ＡＰ１０４からのデータベース接続要求１０６に含まれるユーザを識別する指定をもとに、ユーザ１０３の認証を行う。データ検索部１１３は、データベース１０２に保持しているテーブル１１５のレコードを検索する。アクセス可否判定部１１４は、レコード属性情報１１６に含まれるアクセス権を示す情報を参照し、ＡＰ１０４の問合せ１０７に見合うデータベース処理での、レコードに対するアクセスの可否を判定する。

【００２９】データベース１０２は、レコードを保持するテーブル１１５と、それぞれのレコードに関する属性情報を保持するレコード属性情報１１６からなる。図１では、テーブル１１５のレコードとそのレコードに関する属性情報とが同じ行になるように対応させて図示してある。

【００３０】次に、データベースシステム１０１におけるデータ検索処理の概要を説明する。まず、ＡＰ１０４がデータベース接続要求を行う。この接続要求を、データベース言語SQL文で以下のように記述する。

【００３１】

CONNECT TO dbserver USER 'usr1' … １０６

【００３２】このように、データベース管理システム１０５をデータベースサーバとして識別する名称dbserverを指定し、データベース管理システム１０５がユーザ１０３を識別するために用いるユーザ識別子usr1を指定する。

【００３３】データベース管理システム１０５は、データベース接続要求１０６に従い、ユーザ認証部１１２によりユーザの認証を行い、実行ユーザ情報１１７を作成し、ＡＰ１０４とデータベース管理システム１０５との接続を確立する。

【００３４】次に、ＡＰ１０４が、検索の問合せをデータベース管理システム１０５に送信する。この問合せを、データベース言語SQL文で以下のように記述する。

【００３５】

SELECT clm1 FROM tbl1 WHERE clm2 < 200 … １０７

【００３６】これは、テーブルtbl1から、カラムclm2の値が200より小さいレコードのカラムclm1の値を検索する問合せである。

【００３７】データベース管理システム１０５では、実行制御部１１０による制御のもとで、以下の処理を行う。まず、要求解析部１１１で、問合せ１０７を解析する。解析の結果、要求されたアクションの種別が検索であることを示す要求アクション１１８を作成する。次に、データ検索部１１３が、データベース１０２に保持するテーブル１１５から、問合せの条件を満たすレコードを取得する。次に、アクセス可否判定部１１４が、データ検索部１１３が取得したレコードに対応するレコード属性情報１１６を取得し、該レコード属性情報に含まれるアクセス権を参照して、要求アクションである検索を実行可能かどうかを判定する。

【００３８】図１の例では、実行ユーザ情報がusr1であることから、所有者がusr1のレコードについてはレコード属性情報１１６のうちの所有者アクセス権を、所有者がusr1ではないレコードについてはレコード属性情報１１６のうちの他者アクセス権を参照し、そのアクセス権に検索の実行権を含むレコードについてのみ、検索を実行する。次に、データ検索部１１３は、検索の実行が可能なレコードに対してのみ、SELECT句に指定されたカラムclm1の値を取り出し、結果データ１０９を作成して、問合せ結果１０８としてＡＰ１０４に返却し、検索処理を終了する。

【００３９】この例に示したように、本実施形態によれば以下の効果がある。アクセス可否判定部１１４が、実行ユーザ情報１１７をもとに、レコードのカラム値とは異なるレコード属性情報１１６に含まれるアクセス権を参照してアクセス可否を判定して、データベース管理システム１０５でアクセス制御を行うことから、ＡＰ１０４で複雑なアクセス制御を行うことなく容易にユーザごとのレコード単位のアクセス制御を行うことができる。

【００４０】図２は、図１の実施形態におけるハードウェア構成例を示す図である。本発明の実施形態として示すプログラムは、図２に示すデータ処理装置の上で動作する。

【００４１】データ処理装置２０１－１、２０１－２は、それぞれ、中央演算装置(CPU)２０２－１、２０２－２、主記憶装置（メモリ）２０３－１、２０３－２、入出力(I/O)コントローラ２０４－１、２０４－２、通

信コントローラ２０５－１、２０５－２、およびこれらを接続するシステムバス２０６－１、２０６－２などからなる。また、I/Oコントローラ２０４－１、２０４－２には、キーボードやマウスおよびディスプレイなどのようなデータ入出力装置２０７－１、２０７－２、および、磁気ディスク装置のようなデータ記憶装置２０８－１、２０８－２などが接続される。

【００４２】データ処理装置２０１－１、２０１－２は、通信コントローラ２０５－１、２０５－２によりLAN(Local Area Network)などのネットワーク２０９に接続されており、ネットワーク２０９に接続されているほかのデータ処理装置と通信を行なう。

【００４３】図１およびこれ以降に示すデータ処理は、CPU２０２－１、２０２－２がメモリ２０３－１、２０３－２に格納されたプログラムを実行することにより実現される。AP１０４およびデータベース管理システム１０５の機能を実現するプログラムは、それぞれメモリ２０３－１、２０３－２に格納されて、CPU２０２－１、２０２－２により実行される。なお、AP１０４およびデータベース管理システム１０５は、それぞれソフトウェアの論理的な機能単位であり、それぞれが互いに物理的に異なるデータ処理装置２０１－１、２０１－２上で動作しても良いし、１つのデータ処理装置上でこれらの複数の機能のプログラムが動作しても良い。また、データベース１０２などは、データ記憶装置２０８－１、２０８－２にデータを格納することにより実現される。

【００４４】次に、図１の実施形態のデータベースシステムをさらに詳細に説明する。

【００４５】図３は、図１の実施形態のデータベースシステムの詳細な構成を示す構成図である。基本的な構成は図１に示したシステムの構成と同じであるが、図１で説明したアクセス制御を行うためにインデクスを用いるように具体化している。すなわち、データベース管理システム１０５においてインデクス３０５を用い、データベース管理システム１０５は、インデクス管理部３０１、データ登録部３０２、データ削除部３０３、およびデータ更新部３０４を備えている。

【００４６】インデクス管理部３０１は、一般的なデータベース管理システムにおけるインデクス機能をサポートし、テーブル１１５のレコードのカラム値をキーとし、データベースにおいてレコードを識別する識別子を値とした組をインデクスレコード３０６に保持してインデクス３０５に格納する。インデクス管理部３０１により、キーのカラム値の条件を満たすレコードのレコード識別子を取得する。

【００４７】図４は、テーブル１１５およびレコード属性情報１１６の構造を示す図である。

【００４８】テーブル１１５は、２つのカラムclm1、clm2を備え、tbl1で識別されるテーブルを示す。

【００４９】テーブル１１５は、レコード４０１、４０２、４０３、４０４、４０５を保持する。レコード４０１は、カラムclm1、clm2のそれぞれに対するカラム値aaa、200からなる。レコード４０２は、カラムclm1、clm2のそれぞれに対するカラム値bbb、100からなる。レコード４０３は、カラムclm1、clm2のそれぞれに対するカラム値ccc、100からなる。レコード４０４は、カラムclm1、clm2のそれぞれに対するカラム値ddd、210からなる。レコード４０５は、カラムclm1、clm2のそれぞれに対するカラム値eee、150からなる。

【００５０】レコード属性情報１１６は、テーブル１１５のレコード４０１、４０２、４０３、４０４、４０５のそれぞれに対応する属性情報４０６、４０７、４０８、４０９、４１０を保持する。

【００５１】この属性情報は、所有者、所有者アクセス権、および他者アクセス権からなる。レコード４０１のレコード属性情報４０６の所有者、所有者アクセス権、および他者アクセス権は、それぞれ、usr1、「検索 更新 削除」、および設定無し、であり、これは以下のことを示す。レコード４０１の所有者がusr1で識別されるユーザである。レコード４０１の所有者usr1がレコード４０１に対して行えるデータベース処理のアクションは検索、更新、および削除である。レコード４０１の所有者usr1以外がレコード４０１に対して行えるデータベース処理のアクションは、ない。

【００５２】同様に、レコード４０２に対応するレコード属性情報４０７は、以下のことを示す。レコード４０２の所有者がusr1で識別されるユーザである。レコード４０２の所有者usr1がレコード４０２に対して行えるデータベース処理のアクションは検索、更新、および削除である。レコード４０２の所有者usr1以外がレコード４０２に対して行えるデータベース処理のアクションは検索である。

【００５３】レコード４０３に対応するレコード属性情報４０８は、以下のことを示す。レコード４０３の所有者がusr2で識別されるユーザである。レコード４０３の所有者usr2がレコード４０３に対して行えるデータベース処理のアクションは削除である。レコード４０３の所有者usr2以外がレコード４０３に対して行えるデータベース処理のアクションは検索である。

【００５４】レコード４０４に対応するレコード属性情報４０９は、以下のことを示す。レコード４０４の所有者がusr3で識別されるユーザである。レコード４０４の所有者usr3がレコード４０４に対して行えるデータベース処理のアクションは検索、および削除である。レコード４０４の所有者usr3以外がレコード４０４に対して行えるデータベース処理のアクションは、ない。

【００５５】レコード４０５に対応するレコード属性情報４１０は、以下のことを示す。レコード４０５の所有者がusr2で識別されるユーザである。レコード４０５の所有者usr2がレコード４０５に対して行えるデータベー

ス処理のアクションは検索、更新、および削除である。レコード４０５の所有者usr2以外がレコード４０５に対して行えるデータベース処理のアクションは検索、更新、および削除である。

【００５６】このようなテーブル１１５およびレコード属性情報１１６の構成により、レコード単位のユーザごとのアクセス権を示す情報を保持する。

【００５７】なお、この例ではテーブル１１５とレコード属性情報１１６を分けて構成しているが、レコード属性情報１１６の内容をテーブル１１５のカラムに保持するように、ユーザが定義したカラムとは別にデータベース管理システム１０５が作成して構成しても良い。

【００５８】図５は、図３のインデクス３０５に保持するインデクスレコード３０６の構成を示す図である。図３のインデクス３０５は、図４で説明したtbl1で識別されるテーブル１１５に対し、カラムclm2をキーとして設定されたインデクスである。

【００５９】そのインデクスレコード３０６は、図５に示すように、キー値５０１、レコード識別子５０２、およびレコード属性情報５０３からなる。レコード属性情報５０３は、ユーザ識別子５０４と、所有者アクセス権５０５と、他者アクセス権５０６からなる。図５に図示したインデクスレコード３０６は、テーブル１１５のレコード４０１に対応するインデクスレコードであり、キー値５０１が200、レコード識別子５０２がレコード４０１を識別するレコード識別子rcdid1である。また、レコード属性情報５０３は、レコード４０１の所有者がユーザusr1であり、レコード４０１の所有者アクセス権が検索、更新、および削除であり、レコード４０２の他者アクセス権がなし、であることを示す。

【００６０】このような構成により、インデクスレコード３０６により、ユーザごとのレコード単位のアクセス権を示す。

【００６１】図６は、上述のデータベースシステム１０１におけるデータベース接続処理の流れを示すフローチャートである。

【００６２】まず、AP１０４が、データベース管理システム１０５に対してデータベース接続を要求する（６０１）。すなわち、データベース接続要求１０６をデータベース管理システム１０５に送信する。次に、ユーザ認証部１１２が、データベース接続要求１０６を解析し、ユーザを認証する（６０２）。一般的にデータベース管理システムで行われるユーザ認証で良い。次に、データベース管理システム１０５が、ステップ６０２で認証したユーザに関する情報をもとに実行ユーザ情報１１７を作成する（６０３）。次に、データベース管理システム１０５は、AP１０４との接続を確立する（６０４）。一般的にデータベース管理システムで行われる接続の確立処理で良い。以上でデータベース接続処理を終了する。

【００６３】この処理により、データベース管理システム１０５は、以降に行われるAP１０４の問合せ１０７に従うデータベース処理が、どのユーザからの要求であるかを判別するための実行ユーザ情報１１７を得る。

【００６４】図７は、データベースシステム１０１における基本的なデータベース処理の流れを示すフローチャートである。

【００６５】まず、AP１０４が、データベース管理システム１０５に対して問合せ１０７を送信する（７０１）。次に、データベース管理システム１０５の要求解析部１１１が、問合せ１０７を解析する（７０２）。その解析結果として問合せ解析結果７０５を出力する。また、問合せで要求されている検索、更新、あるいは削除などのデータベース処理のアクションを示す要求アクション１１８を出力する。

【００６６】次に、データベース管理システム１０５が問合せ１０７に見合うデータベース処理を行う（７０３）。すなわち、要求アクション１１８に示される検索、更新、削除などの要求に応じて、実行ユーザ情報１１７をもとに、アクセス可否判定部１１４によりアクションの実行可否を判定しながら、データベース処理を行う。この詳細は、後に図８、図１４、図１６、および図２１などを用いて説明する。データベース処理の結果、問合せ処理結果１０８を出力する。

【００６７】データベース管理システム１０５は、問合せ処理結果１０８をAP１０４に返却し（７０４）、データベース処理を終了する。

【００６８】次に、データベースシステム１０１における検索処理の詳細について説明する。AP１０４の検索要求は、図１に示した問合せ１０７と同じとする。また、全体的なデータベース処理の流れは、図７に示したとおりであり、検索処理の詳細については、図７のステップ７０３のデータベース管理システム１０５における処理の流れとして説明する。

【００６９】図８は、データベース管理システム１０５における検索処理のフローチャートである。

【００７０】まず、データベース管理システム１０５は、問合せ解析結果７０５をもとに検索条件８０５を取得する（８０１）。例えば、図１に示した問合せ１０７であれば、clm2 < 200 という検索条件を得る。

【００７１】次に、データ検索部１１３が、データベース１０２のレコードから結果データ８０６を作成する（８０２）。具体的には、データ検索部１１３は、検索アクションであることを示す要求アクション１１８、および要求したユーザがusr1であることを示す実行ユーザ情報１１７をもとに、アクセス可否判定部１１４が実行可否を判定したレコードを処理対象として、検索処理を行う。検索処理結果として結果データ８０６を出力する。詳細は後に図９を用いて説明する。

【００７２】次に、データベース管理システム１０５は、ステップ８０２で結果データ８０６が得られたかど

うかを判定する（８０３）。結果データ８０６がある場合は、該結果データを問合せ処理結果１０８に設定し（８０４）、ステップ８０２に戻って、次の検索結果データを作成するよう処理を繰り返す。ステップ８０３の判定で結果データがない場合（データベースの該当するレコードをすべて処理した場合）は、この処理を終了する。

【００７３】図９は、データ検索部１１３における検索処理結果作成の処理（図８のステップ８０２）を示すフローチャートである。

【００７４】まず、データ検索部１１３は、検索処理対象にインデクスが設定されているか否かを判定する（９０１）。

【００７５】インデクスが設定されている場合は、インデクス管理部３０１が問合せの条件を満たしかつアクセス可能なレコードのレコード識別子９０５を取得し（９０２）（この処理の詳細は後に図１０を用いて説明する）、続いてそのレコード識別子９０５をもとにデータ検索部１１３がデータベース１０２からレコードを取得して結果データ８０６を作成し（９０３）、この検索処理を終了する。

【００７６】ステップ９０１の判定でインデクスが設定されていない場合は、データ検索部１１３がデータベース１０２から問合せの条件を満たしかつアクセス可能なレコードを取得して結果データ８０６を作成し（９０４）（この処理の詳細は後に図１２を用いて説明する）、この検索処理を終了する。

【００７７】図１０は、インデクス管理部３０１における検索処理（図９のステップ９０２）のフローチャートである。

【００７８】まず、インデクス管理部３０１は、検索条件８０５をもとに、条件を満たすインデクスレコード１００６を取得する（１００１）。

【００７９】次に、アクセス可否判定部１１４において、インデクスレコード１００６に対応するレコード属性情報および検索要求であることを示す要求アクション１１８をもとに、要求アクションについての実行可否を判定する（１００２）。アクセス可否判定部１１４による実行可否の判定処理については、図１１で説明する。

【００８０】次に、アクセス可否判定部１１４の判定結果が実行可であるかどうかを判定する（１００３）。実行可であれば、インデクスレコード１００６に含まれるレコード識別子９０５を出力し（１００４）、この処理を終了する。ステップ１００３の判定で実行可でない場合は、「該当するレコードなし」とし（１００５）、この処理を終了する。

【００８１】このような処理により、データベース管理システム１０５は、実行ユーザの情報とインデクスに含まれるレコード属性情報のアクセス権情報をもとにして、ユーザからのデータベース処理要求について実行ユーザごとのレコード単位のアクセス制御を行うことができる。

【００８２】図１１は、アクセス可否判定部１１４における要求アクション実行可否を判定する処理（図１０のステップ１００２）のフローチャートである。

【００８３】まず、アクセス判定部１１４は、この処理の入力である実行ユーザ情報１１７とレコード属性情報１１０７をもとに、実行ユーザが当該レコードの所有者であるか否かを判定する（１１０１）。実行ユーザがレコードの所有者の場合は、レコード属性情報１１０７から所有者のアクセス権を示す情報をアクセス権情報１１０８として取得し（１１０２）、ステップ１１０４にすすむ。ステップ１１０１で実行ユーザがレコードの所有者と異なる場合は、レコード属性情報１１０７から他者のアクセス権を示す情報をアクセス権情報１１０８として取得し（１１０３）、ステップ１１０４にすすむ。

【００８４】次に、ステップ１１０４では、ステップ１１０２または１１０３で取得したアクセス権情報１１０８とこの処理の入力である要求アクション１１０９をもとに、要求アクションの実行権があるか否かを判定する。例えば、要求アクションが検索で、アクセス権情報に検索が含まれる場合は、実行権ありとする。ステップ１１０４の判定で実行権ありの場合は、実行可とし（１１０５）、この処理を終了する。ステップ１１０４の判定で実行権なしの場合は、実行不可とし（１１０６）、この処理を終了する。

【００８５】このような処理により、データベース管理システム１０５は、実行ユーザの情報とレコード属性情報に含まれるアクセス権情報をもとにして、ユーザからのデータベース処理要求について実行ユーザごとのレコード単位のアクセス制御を行うことができる。

【００８６】図１２は、データ検索部１１３における検索処理のフローチャートである。この処理は、図９のステップ９０４の処理の詳細を示す。

【００８７】まず、データ検索部１１３で、データベース１０２から、問合せの条件８０５を満たすレコードを探索する（１２０１）。また、探索して得られるレコードから、それに対応するレコード属性情報１２０８を取得する。

【００８８】次に、ステップ１２０１で条件を満たすレコードが得られたか否かを判定する（１２０２）。レコードが得られた場合は、要求アクション１１８および実行ユーザ情報１１７をもとに、アクセス可否判定部１１４において検索の要求アクションが実行可であるか否かを判定する（１２０３）。この判定処理は、図１１に示した処理で行う。

【００８９】続いて、ステップ１２０３の判定結果が実行可であるかどうかを判定する（１２０４）。実行可の場合は、ステップ１２０１で得られたレコードをもとに結果データ８０６を作成し（１２０５）、この処理を終

了する。ステップ１２０４で実行可でない場合は、「該当結果なし」とし（1206）、この処理を終了する。

【００９０】ステップ１２０２でレコードがなかった場合は、「該当結果なし」とし（１２０７）、この処理を終了する。

【００９１】このような処理により、データベース管理システム１０５は、実行ユーザの情報とデータベースに含まれるレコード属性情報のアクセス権情報をもとにして、ユーザからのデータベース処理要求について実行ユーザごとのレコード単位のアクセス制御を行うことができる。

【００９２】次に、本実施形態のデータベースシステム１０１におけるデータ登録処理の詳細を説明する。

【００９３】図１３は、本実施形態のデータベースシステム１０１の構成を示す図である。基本的には、図３に示したデータベースシステム１０１と同様の構成であるが、データの登録の流れに着目した構成を示してある。すなわち、AP１０４が、登録するデータ１３０３に対してデータベース１０２におけるアクセス権をどのようにするかを指定するアクセス権設定要求１３０１と、データ登録要求の問合せ１３０２を発行すること、並びに、データベース管理システム１０５において、要求アクション１１８が「登録」を示すこと、およびアクセス権設定要求１３０１に応じてアクセス権情報１３０４を持つことなどを示してある。

【００９４】このアクセス権設定要求１３０１

SET INSERT_DATA_PERMISSION 'SUD---'

は、以降に登録するデータについて、所有者のアクセス権が「検索 更新 削除」であり、他者のアクセス権がなし、とすることを示す。要求解析部１１１は、このようなアクセス権設定要求１３０１を解析し、アクセス権情報１３０４を得る。

【００９５】登録要求の問合せ１３０２

INSERT INTO tbl1 VALUES ('fff' 300)

は、テーブルtbl1に登録データ１３０３（その値は、fff、300）からなるレコードを登録することを示す。

【００９６】次に、データベースシステム１０１でのデータ登録処理の詳細を説明する。全体的なデータベース処理の流れは、図７に示したとおりであり、登録処理の詳細については、図７のステップ７０３のデータベース管理システム１０５における処理の流れとして説明する。

【００９７】図１４は、データベース管理システム１０５における登録処理のフローチャートである。

【００９８】まず、データベース管理システム１０５が、登録データ１３０３をもとに、データベース１０２に格納するレコードを作成する（１４０１）。図１３の登録データ１３０３の例であれば、カラムclm1、clm2に対しそれぞれfff、300のカラム値からなるレコードを作成する。

【００９９】次に、データベース管理システム１０５が、実行ユーザ情報１１７およびアクセス権情報１３０４をもとに、レコード属性情報を作成する（１４０２）。実行ユーザ情報１１７に含まれるユーザ識別子usr1と、アクセス権情報１３０４の所有者アクセス権「検索 更新 削除」、他者アクセス権「なし」をもとに、レコード属性情報を作成する。

【０１００】次に、データ登録部３０２が、データベース１０２中に、ステップ１４０１および１４０２で作成したレコードとレコード属性情報とを対応付けて格納する（１４０３）。格納した結果、格納したレコードを識別するレコード識別子１４０６を作成する。

【０１０１】次に、登録したレコードを保持するテーブル１１５に対してインデクスが設定されているか否かを判定する（１４０４）。インデクスが設定されている場合は、インデクス管理部３０１においてインデクスレコード登録処理を行い（１４０５）（インデクスレコード登録処理の詳細は図１５で説明する）、この処理を終了する。ステップ１４０４でインデクスが設定されていない場合は、この処理を終了する。

【０１０２】このような処理により、AP１０４が登録要求の問合せでユーザの情報やアクセス権に関わる情報を指定しなくても、データベース管理部１０５がレコード属性情報１１６にアクセス制御に必要な情報を設定するので、AP１０４に負担をかけることなく、アクセス制御することができる。

【０１０３】図１５は、インデクス管理部３０１におけるインデクスレコード登録処理のフローチャートである。これは図１４のステップ１４０５の処理の詳細を示す。

【０１０４】まず、インデクス管理部３０１は、AP１０４からの登録要求で指定された登録データ１３０３と、ステップ１４０３で格納して得られたレコード識別子（本例では、rcdid2）１４０６をもとに、インデクスに登録するインデクスレコード１５０４を作成する（１５０１）。本例では、カラムclm2のキー値300と、レコードを識別するレコード識別子rcdid2を含むインデクスレコード１５０４を作成することになる。

【０１０５】次に、実行ユーザ情報１１７とアクセス権情報１３０４をもとに、レコード属性情報１５０５を作成する（１５０２）。本例では、実行ユーザusr1と、アクセス権情報 'SUD---' からなるレコード属性情報を作成することになる。次に、インデクス３０５にインデクスレコード１５０４とレコード属性情報１５０５とを対応付けて登録し（１５０３）、この処理を終了する。

【０１０６】このような処理により、データベース管理部１０５がインデクス３０５のインデクスレコード３０６のレコード属性情報にアクセス制御に必要な情報を設定するので、AP１０４に負担をかけることなく、アクセス制御することができる。

【0107】次に、データベースシステム101において、テーブル115に保持するレコードに同一のレコードが存在しないようにUNIQUE指定した場合の、UNIQUEチェックを含むデータ登録処理の詳細を説明する。なお、この例でのUNIQUEチェックでは、ユーザごとにレコードの同一性がチェックされるものとする。すなわち、各ユーザごとに所有するレコードがそれぞれユニークになり、登録するユーザが異なる（レコードの所有者が異なる）場合は、同じテーブルに同一のカラム値からなるレコードが存在しうるものとする。

【0108】全体的なデータベース処理の流れは、図7に示したとおりであり、本登録処理の詳細については、図7のステップ703のデータベース管理システム105における処理の流れとして説明する。

【0109】図16は、データベース管理システム105におけるUNIQUEチェックを含むデータ登録処理のフローチャートである。基本的な処理の流れは、図14に示したものと同じである。異なる部分のみを説明する。

【0110】まず、ステップ1401、1402で登録するレコードおよびレコード属性情報を作成したのち、ステップ1403のデータベース102へのレコード格納の前に、UNIQUEチェックを行う（1601）。チェック処理の詳細については、図17で説明する。

【0111】続いて、UNIQUEチェックの結果、すでにレコードが登録されているか否かを判定する（1602）。レコード登録がある場合は、「登録済み」エラーとし（1603）、この処理を終了する。ステップ1602でレコードが登録されていない場合は、ステップ1403以降の図14で説明したとおりの処理を行う。

【0112】図17は、データベース管理システム105でのUNIQUEチェック処理（図16のステップ1601）のフローチャートである。

【0113】まず、データを登録するテーブル115にインデクスが設定されているか否かを判定する（1701）。インデクスが設定されている場合は、インデクス管理部301が、登録データ1303と実行ユーザ情報117をもとに、登録データと同一視するレコードを探索する（1702）。このステップの詳細については、図18で説明する。

【0114】続いて、ステップ1702での探索の結果、インデクスに登録があるか否かを判定する（1703）。登録がある場合は、「すでに登録あり」とし（1704）、この処理を終了する。登録がない場合は、「登録なし」とし（1705）、この処理を終了する。

【0115】ステップ1701で、インデクスが設定されていない場合は、登録データ1303と実行ユーザ情報117をもとに、データ検索部113がデータベース102から登録データと同一視するレコードを探索する（1706）。このステップの詳細は図19で説明する。

【0116】続いて、ステップ1706での探索の結果、すでにデータベース102にレコードが登録されているか否かを判定する（1707）。登録がある場合は、「すでに登録あり」とし（1708）、この処理を終了する。登録がない場合は、「登録なし」とし（1709）、この処理を終了する。

【0117】図18は、インデクス管理部301が登録データと同一視するレコードを探索する処理（図17のステップ1702）のフローチャートである。

【0118】まず、インデクス305において、キーが登録データ1303のキーと一致するインデクスレコード1808を探索する（1801）。次に、ステップ1801での探索の結果、一致するインデクスレコードがあるか否か判別する（1802）。判別の結果、インデクスレコードがない場合は、「登録なし」とし（1803）、この処理を終了する。

【0119】ステップ1802でインデクスレコードがある場合は、インデクスレコード1808に対応するレコード属性情報1809を取得する（1804）。次に、実行ユーザ情報117とレコード属性情報1809に含まれる所有者をもとに、実行ユーザとレコードの所有者が一致するか否かを判定する（1805）。ユーザが一致する場合は、「すでに登録あり」とし（1806）、この処理を終了する。ユーザが一致しない場合は、「登録なし」とし（1807）、この処理を終了する。

【0120】このような処理により、データベース管理システム105が、インデクスを用いて、ユーザごとのレコードのユニークチェックを行うことができる。

【0121】図19は、データ検索部113でデータベース102から登録データと同一視するレコードを探索する処理（図17のステップ1706）のフローチャートである。

【0122】まず、データ検索部113が、データベース102から登録データ1303と一致するレコードを探索する（1901）。次に、ステップ1901の探索の結果、一致するレコードがあったか否かを判定する（1902）。

【0123】一致するレコードがあった場合は、そのレコードに対応するレコード属性情報1908を取得し（1903）、実行ユーザ情報117とレコード属性情報1908の所有者とが一致するか否か判定する（1904）。一致する場合は、「すでに登録あり」とし（1905）、この処理を終了する。一致しない場合は、「登録なし」とし（1906）、この処理を終了する。

【0124】ステップ1902でデータベースに一致するレコードがない場合は、「登録なし」とし（1906）、この処理を終了する。

【0125】このような処理により、データベース管理システム105が、データベース102に保持したレコ

ード属性情報１１６をもとにして、ユーザごとのレコードのユニークチェックを行うことができる。

【０１２６】なお、上記の処理では、レコードを構成するカラム値のほかに、レコードに関する属性情報を含めて、レコードの同一性を判定している。

【０１２７】次に、データベースシステム１０１におけるデータベース１０２のデータ削除処理の詳細を説明する。全体的なデータベース処理の流れは、図７に示したとおりであり、削除処理の詳細については、図７のステップ７０３のデータベース管理システムにおける処理の流れとして説明する。

【０１２８】図２０は、AP１０４におけるデータ削除要求の問合せ１０７である。この問合せ１０７のSQL文

【０１２９】DELETE FROM tbl1 WHERE clm2 = 200

は、テーブルtbl1から、カラムclm2の値が200であるレコードを削除することを示す。

【０１３０】図２１は、データベース管理システム１０５でのデータ削除処理のフローチャートである。

【０１３１】まず、問合せ要求を解析し、削除対象の条件２１０５を取得する（２１０１）。例えば、図２０に示したような削除要求を要求解析部１１１で解析し、その解析結果である問合せ解析結果７０５から、条件「テーブルtbl1のカラムclm2の値が200であるレコード」という条件を取得する。

【０１３２】次に、削除を示す要求アクション１１８および実行ユーザ情報１１７をもとに、データ検索部１１３でユーザが削除可能なデータベース１０２中のレコードを探索する（２１０２）。このステップの詳細は、図９に示したデータ検索処理と同様で、異なる点は、要求アクション１１８を「削除」とすることのみであるので、説明は省略する。

【０１３３】次に、ステップ２１０２で探索した結果、レコードがあるかどうかを判定する（２１０３）。レコードがある場合は、データ削除部３０３が該当するレコードを削除し（２１０４）、ステップ２１０２に戻って処理を繰り返す。ステップ２１０３でレコードがない場合（削除対象をすべて削除した場合）は、この処理を終了する。

【０１３４】このような処理により、AP１０４が複雑な問合せ要求をしなくても、データベース管理システム１０５が、データ削除処理においてユーザごとにレコード単位でアクセス制御することができる。

【０１３５】図２２は、図２１に示した処理により削除されるレコードの構成例を示す図である。例えば、図２０に示した問合せ１０７の削除対象条件が clm2 = 200 であることから、レコード２２０１、２２０２、２２０３がデータベース１０２から探索されるが、アクセス可否判定部１１４により、レコード属性情報１１６に従い、この削除の実行ユーザusr1のアクセス権が「削除」を含むかどうかが判定される。

【０１３６】レコード２２０１は、所有者が実行ユーザと同じusr1で、所有者アクセス権に削除が含まれるので、削除対象となる。レコード２２０２は、所有者が実行ユーザと異なるusr2で、他者アクセス権に削除が含まれるので、削除対象となる。レコード２２０３は、所有者が実行ユーザと異なるusr3で、他者アクセス権に削除が含まれないので、削除対象とならない。

【０１３７】次に、データベースシステム１０１で、ユーザ削除と同時にそのユーザの所有するデータをすべて削除する例を示す。この例のデータベースシステムの構成は、図３に示したものと同じである。

【０１３８】図２３は、AP１０４において、ユーザ削除と同時にそのユーザの所有するデータをすべて削除する要求の例を示す。２３０１に示す設定文

SET DELETE_USER_DATA_AT_DROP_USER

によって、データベース管理システム１０５に、ユーザ削除と同時にそのユーザの所有するデータをすべて削除するよう設定する。続いて、ユーザ削除要求２３０２

DROP USER usr1

により、データベース１０２からユーザusr1を削除することを要求する。

【０１３９】データベース管理システム１０５は、ユーザ削除要求２３０２を受け付けると、２３０１の設定に従い、データベース１０２内のすべてのテーブルについて、レコード属性情報の所有者がusr1のデータをすべて削除する。削除処理は、図２１に示した処理とほぼ同じであるが、データベース管理システム１０５のユーザ削除処理に伴う特権により、所有者がusr1であればアクセス権に関係なくレコードを削除する。所有者usr1のレコードをすべて削除したのち、ユーザusr1の登録を抹消し、ユーザ削除処理を終える。

【０１４０】次に、データベースシステム１０１で、ロールごとにアクセス制御する例を説明する。ロールとは、データベースで同一の特権を持つユーザの集まりを示す。この例のデータベースシステム１０５の構成は、図３に示したものと同じである。異なる点は、レコード属性情報１１６に、レコードの属するロール、およびそのロールに属するユーザのアクセス権を持つことである。

【０１４１】図２４に、ロールを含むレコード属性情報１１６の例を示す。２４０３は、レコード２４０１のロールがrole1であり、ロールrole1のユーザからのアクセス権が検索と更新であることを示す。２４０４は、レコード２４０２のロールがrole2であり、ロールrole2のユーザからのアクセス権が検索のみであることを示す。このようなレコード属性情報１１６に従い、前述のようなアクセス可否判定部１１４による判定を行い、ロールごとのアクセス制御を行うことができる。

【０１４２】次に、データベースシステム１０１で、アクセス制御を行うモジュールを、データベース管理シス

テム105とは独立したモジュールとして構成する例を示す。

【0143】図25に、アクセス制御を行うモジュールを独立させたデータベース管理システムの構成図を示す。この例のデータベース管理システム105の基本的な構成は、図3に示したものと同じである。異なる点は、アクセス可否判定部114およびインデクス管理部301を、データベース管理システム105自体に埋め込んで固定的に備えるのではなく、独立したモジュールとしていることである。

【0144】アクセス可否判定部114およびインデクス管理部301のモジュールは、データベース管理システム105に対して、いわゆるプラグイン（plug-in）の機構により取り込まれるものとする。例えば、オペレーティングシステムの動的ロード機能を用いて、データベース管理システム105が、アクセス可否判定部114およびインデクス管理部301のモジュールの実体である動的ロードライブラリをロードして取り込むものとする。

【0145】この構成では、テーブルを次のようなSQL文で定義する。

【0146】CREATE TABLE tbl1 (...)
WITH OPTIONS (ACCESS CONTROL LIBRARY 'libactl1')

【0147】この定義文は、テーブルtbl1についてアクセス制御を行うことを指定し、ライブラリlibactl1によって提供される機能を用いてアクセス制御を行うことを示す。

【0148】データベース管理システム105の実行制御部110は、この定義の指定に従い、データベース処理においてアクセス制御が必要なタイミングでライブラリをロードして処理を行う。例えば、図12のステップ1203において、ライブラリlibactl1に含まれるアクセス可否を判定する機能を呼び出す（関数名checkAccessPermission()の関数をコールする）。

【0149】また、インデクスを次のようなSQL文で定義する。

【0150】CREATE INDEX idx1 ON tbl1 (clm1)
WITH OPTIONS (ACCESS CONTROL LIBRARY 'libactlidx1')

【0151】この定義文は、テーブルtbl1のカラムclm1に設定したインデクスidx1についてアクセス制御を行うことを指定し、ライブラリlibactlidx1によって提供される機能を用いてアクセス制御を行うことを示す。

【0152】データベース管理システム105の実行制御部110は、この定義の指定に従い、データベース処理において、インデクスを利用してアクセス制御が必要なタイミングでライブラリのロードして処理を行う。例えば、図10のステップ1002において、ライブラリlibactlidx1に含まれるアクセス可否を判定する機能を＊呼び出す（関数名checkAccessPermissionByIndexEntry()の関数をコールする）。

【0153】また、アクセス可否判定を行うライブラリlibactl1は、アクセス権に関する情報（レコード属性情報116）を操作する機能を持ち、データベース管理システム105がテーブル115のレコードを登録および削除するタイミングにおいて、それぞれの機能（関数insertPermissionRecord(),deletePermissionRecord()）を呼び出すことにより、レコードの登録および削除に合わせてレコード属性情報116を設定する。

【0154】インデクス管理を行うライブラリlibactlidx1は、アクセス権に関する情報（インデクスレコード306のレコード属性情報503）を操作する機能を持ち、データベース管理システム105がテーブル115のレコードを登録405および削除するタイミングにおいて、それぞれの機能（関数insertPermissionIndexEntry(),removePermissionIndexEntry()）を呼び出すことにより、レコードの登録および削除に合わせてレコード属性情報503を設定する。

【0155】このように、アクセス制御を行うモジュールをデータベース管理システム105と独立させることにより、データベース管理システム105にアクセス制御の機能を追加、削除することを容易に行うことができる。また、セキュリティ面では、データベース管理システム105自体を詳細に解析して攻撃が可能となった場合でも、アクセス制御の部分が不明で攻撃が不可能となる場合もあり、セキュリティを強化することができる。

【0156】上記実施の形態において、レコード単位に限らず、カラム単位に属性情報を対応付けることにより、カラム値単位のアクセス制御を行うこともできる。

【0157】次に、データベースシステム101で、カラム値単位でアクセス制御を行うモジュールを、データベース管理システム105とは独立したモジュールとして構成する例を示す。

【0158】図26に、カラム値単位でアクセス制御を行うモジュールをとり込むデータベース管理システムの構成図を示す。この例のデータベース管理システム105の基本的な構成は、図3に示したものと同じである。異なる点は、国際標準化機構ISOの標準SQL99で規定されている抽象データ型を用いて、カラム値単位でアクセス制御を行うモジュールである抽象データアクセス管理部2601をデータベース管理システム105に取り込んでいることである。また、抽象データ型のカラムに対するインデクス機能を提供する抽象データインデクス管理部2602もデータベース管理システム105に取り込んでいる。

【0159】ここで、抽象データ型は次のようなSQL文で定義する。

【0160】

CREATE TYPE adt1

```
( PRIVATE attr1 CHAR(200),
  PRIVATE permissions VARCHAR(100),
  PUBLIC FUNCTION getattr1( ) RETURNS CHAR(200)
  LIBRARY 'libadt1' ...
)
```

【０１６１】関数getattr1()の実装は、抽象データアクセス制御部２６０１のライブラリlibadt1に含まれる。

【０１６２】抽象データ型adt1をカラムの型とするテーブルは、次のようなSQL文で定義する。

【０１６３】

```
CREATE TABLE tbl1 ( ..., clm3 adt1, ... )
```

【０１６４】このような定義により、抽象データ型の値をカラム値とする。

【０１６５】図２７に、アクセス制御を行う抽象データ型のカラム値のデータ構造を示す。抽象データ型の値２７０１は、属性値２７０２とアクセス権情報２７０３からなる。属性値２７０２は、先述の抽象データ型adt1の定義文での attr1 CHAR(200)に従い、200バイトの文字配列の値を保持する。

【０１６６】アクセス権情報２７０３は、この抽象データ型の値２７０１へのアクセス可否についての情報を保持する。アクセス権情報２７０３は、先述の抽象データ型adt1の定義文での permissions VARCHAR(100) に従い、最大100バイトの可変長文字列値を保持する。詳細は、図４のレコード属性情報１１６と同様である。

【０１６７】関数getattr1()では、アクセス権情報２７０３を参照し、実行ユーザ情報１１７をもとに、アクセス可否を判定し、アクセス可と判定した場合は属性値２７０２を返し、アクセス否と判定した場合は、「該当なし」を返すものとする。判定処理は、図１１で説明した処理と同様でよい。

【０１６８】抽象データ型のカラム値の参照は、次のようなSQL文で要求する。

【０１６９】SELECT clm3.getattr1() FROM tbl1

【０１７０】このSQL文により、テーブルtbl1のカラムclm3に保持した抽象データ型の属性attr1の属性値を取得する。関数getattr()のアクセス可否判定により、アクセス権情報２７０３によりアクセス可であることが示された抽象データ型の値の属性値のみが問合せ結果となる。

【０１７１】抽象データインデクス管理部２６０２は、抽象データ型のカラムに対して、インデクス３０５と同様に、抽象データ型のカラム値をキーとしたインデクスを作成する。そして、図１０、図１５に示した処理と同様に、アクセス可否判定を含むインデクス処理を行う。データベース処理のそれぞれのタイミング、インデクス作成、エントリ検索、エントリ登録、エントリ削除について、それぞれ抽象データインデクス管理部２６０２に含まれる関数createIndex(), searchEntry(),insertEntry(), removeEntry()をコールし、アクセス制御に関する処理を抽象データインデクス管理部２６０２で行う。

【０１７２】このような構成により、カラム値ごとにアクセス制御を実装できるので、カラムごとにアクセス制御機能を追加、削除することが容易になる。また、個々のカラムごとに抽象データ型の関数を実装することにより、きめこまかなアクセス制御が可能になる。また、抽象データインデクス管理部２６０２により、アクセス制御を含む抽象データ型のカラム値に対する問合せ処理を高速化することができる。

【０１７３】上記実施形態において、レコード属性情報１１６の所有者の情報に基づき、テーブルの結合（Join）において、所有者が一致するレコード同士を結合することもできる。レコード属性情報１１６を用いることにより、AP１０４が複数のテーブルでユーザを示すカラムやユーザ識別子の整合性を維持する手間が不要となる。

【０１７４】また、データベース１０２のレコードを暗号化して格納する場合に、レコード属性情報１１６の所有者に従って、ユーザごとに暗号化方式を変更することができ、ユーザの区別なく暗号化する場合に比べてセキュリティを強化することができる。

【０１７５】また、レコード属性情報１１６の所有者に従って、データベース１０２に格納するレコードをユーザごとにクラスタリングすることにより、データ記憶装置２０８－２からのレコードの読み込みの効率を向上し、ユーザごとのデータベース処理の高速化を図ることができる。

【０１７６】また、レコードごとの所有者とアクセス権の管理を、データベース管理システム１０５がオペレーティングシステムのユーザおよびアクセス管理に適合させ、またアプリケーションプログラムにもユーザおよびアクセス管理の情報を引き渡すことにより、オペレーティングシステム、データベース、およびアプリケーションプログラムを統一したアクセス制御を行うことができる。このことは、特に一貫したセキュリティが要求されるデータ管理システムにおいて有効である。

【０１７７】また、本発明によるアクセス制御に類似した方法として、一般的なファイルシステムにおけるファイルのパーミッションによるアクセス制御方法があるが、そのようなファイルシステムではファイルの存在が他ユーザにも認識され、アクセス時にパーミッションがない場合にエラーとなるのに対し、本発明の方法では他ユーザにはレコードの存在すら認識されないようにすることができる。また、ファイルシステムではファイル名がファイルの識別に用いられユーザが異なっても同一の

名称のファイルを作成することができないが、本発明の方法では所有者情報を含めた同一性の判定によって、レコード自体は同一でもユーザごとにレコードを登録することができる。このように、自分のレコードを他ユーザに存在すら認識させないようにできることは、セキュリティが要求されるデータ管理システムにおいて有効である。

【0178】なお、前述したフローチャートの処理は、図2に示したようなデータ処理装置でプログラムを実行することによって実現できる。また、そのプログラムは、ハードディスク装置、フロッピーディスクなどのコンピュータで読み書きができる記憶媒体に格納することができ、ネットワークを通してプログラムにアクセスすることができる。

【0179】

【発明の効果】以上説明したように、本発明によれば、複数のユーザのデータを管理し、複数のユーザがデータにアクセスするデータ管理システムにおいて、データベースを利用するアプリケーションプログラムが意識しなくても、データベース管理システムがユーザごとのレコード単位のアクセス制御を行うことができるので、アプリケーションプログラムを簡略化できるという効果がある。

【図面の簡単な説明】

【図1】本発明の一実施形態におけるデータ検索処理の概要を示す概念図

【図2】本発明の一実施形態におけるハードウェアの構成図

【図3】本発明の一実施形態におけるデータベースシステムの構成図

【図4】データベースのテーブルの構成図

【図5】データベースのインデクスレコードの構成図

【図6】データベース接続処理のフローチャート図

【図7】データベース問合せ処理のフローチャート図

【図8】データベース管理システムにおけるデータ検索処理のフローチャート図

【図9】データ検索部におけるデータ検索処理結果作成のフローチャート図

【図10】インデクス管理部におけるデータ検索処理のフローチャート図

【図11】アクセス可否判定処理のフローチャート図

【図12】データ検索部におけるデータ検索処理のフローチャート図

【図13】本発明の一実施形態のデータベースシステムの構成図

【図14】データベース管理システムにおけるデータ登録処理のフローチャート図

【図15】インデクス管理部におけるインデクスレコード登録処理のフローチャート図

【図16】データベース管理システムにおけるUNIQUEチェックを含むデータ登録処理のフローチャート図

【図17】データベース管理システムにおけるUNIQUEチェック処理のフローチャート図

【図18】インデクス管理部におけるUNIQUEチェック処理のフローチャート図

【図19】データ検索部におけるUNIQUEチェック処理のフローチャート図

【図20】データ削除要求の構成図

【図21】データベース管理システムにおけるデータ削除処理のフローチャート図

【図22】データ削除対象レコードの構成図

【図23】ユーザ削除と同時にデータ削除する要求の構成図

【図24】ロールの情報を含むテーブルの構成図

【図25】アクセス制御を行うモジュールを独立させたデータベース管理システムの構成図

【図26】抽象データ型のカラム値のアクセス制御を行うモジュールを用いたデータベース管理システムの構成図

【図27】抽象データ型のカラム値の構成図

【符号の説明】

105…データベース管理システム

112…ユーザ認証部

114…アクセス可否判定部

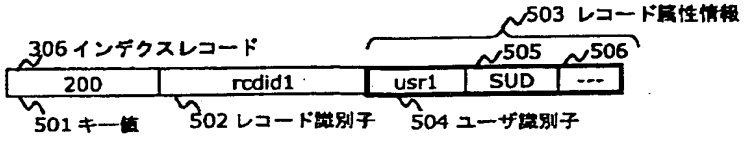
116…レコード属性情報

117…実行ユーザ情報

118…要求アクション

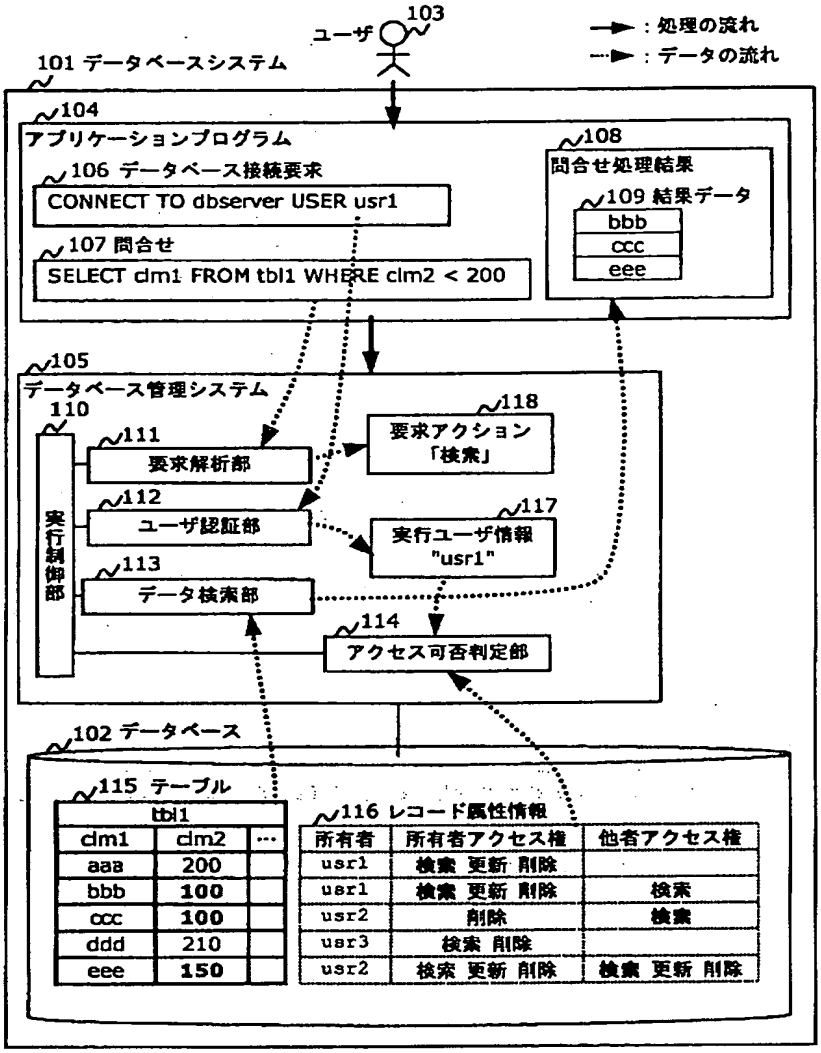
【図5】

インデクスレコードの構成図（図5）



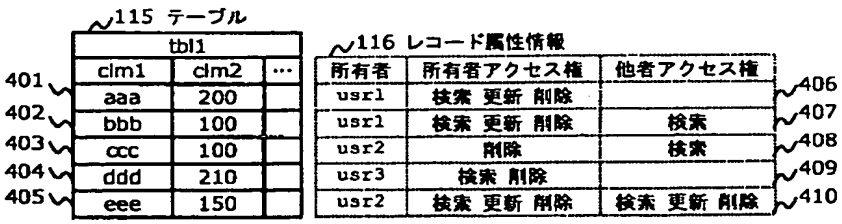
【図1】

データ検索処理の概念図（図1）



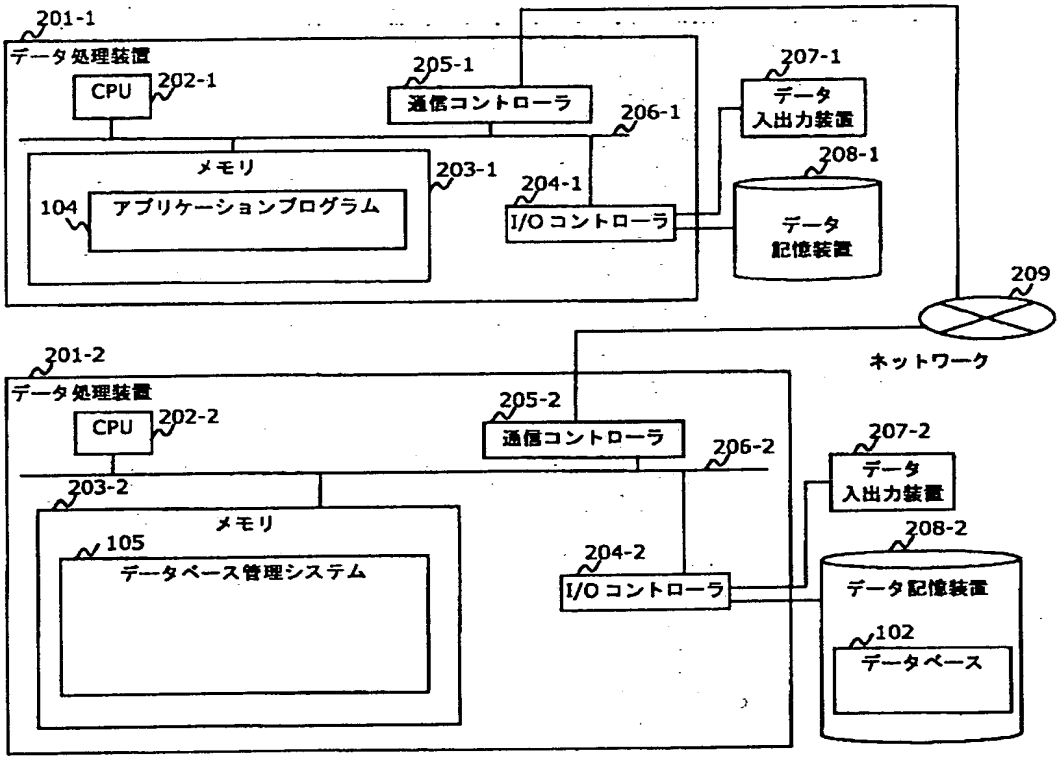
【図4】

データベースのテーブルの構成図（図4）



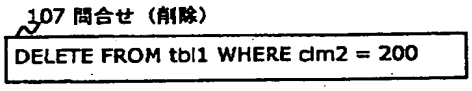
【図2】

ハードウェア構成図（図2）



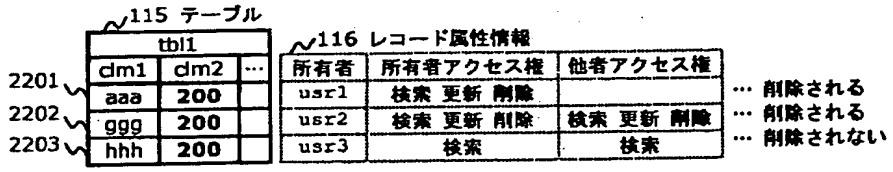
【図20】

データ削除要求の構成図（図20）



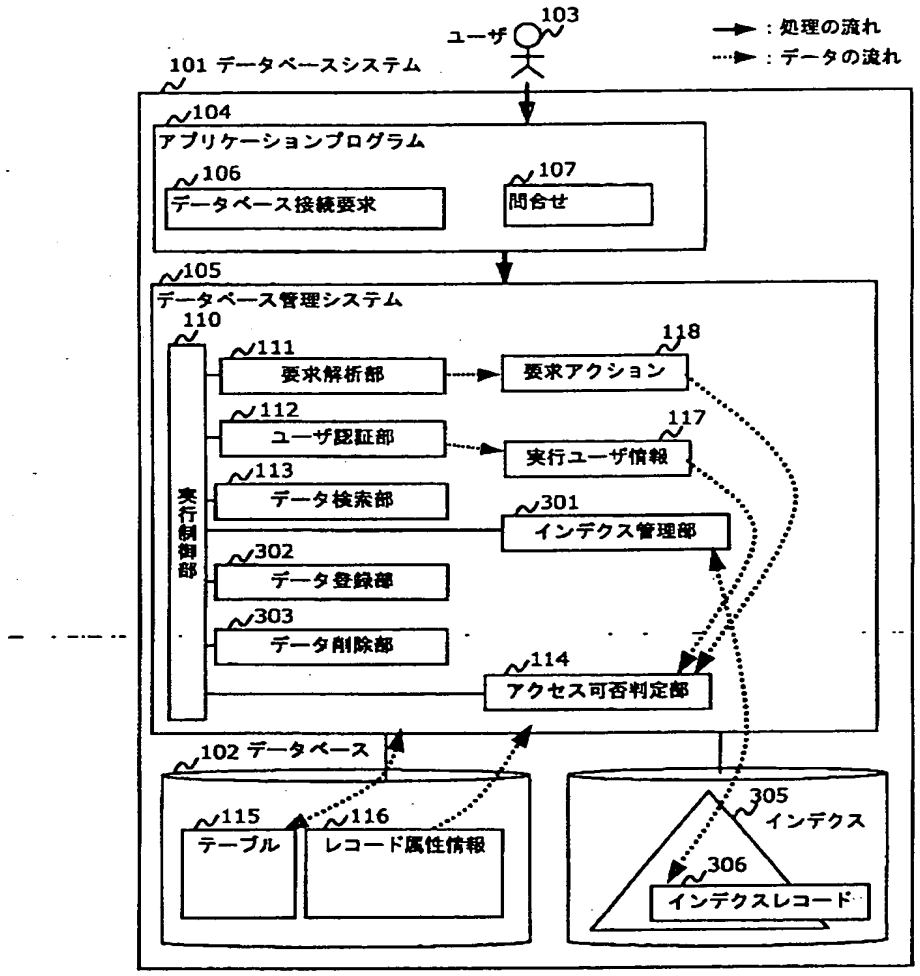
【図22】

削除対象のレコードの構成図（図22）



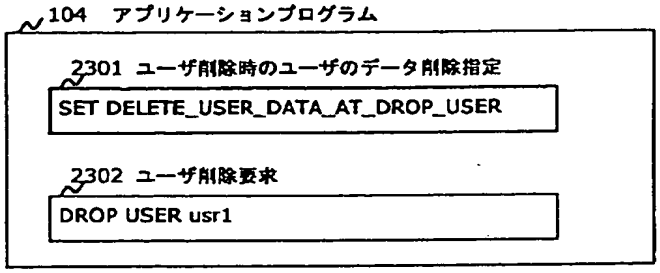
【図3】

データベースシステムの構成図（図3）



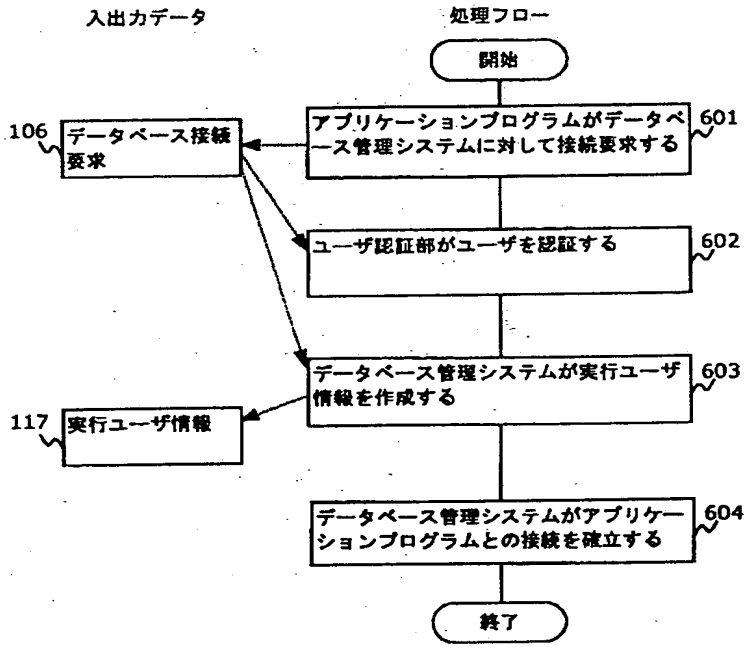
【図23】

ユーザ削除と同時にデータ削除する要求の構成図（図23）



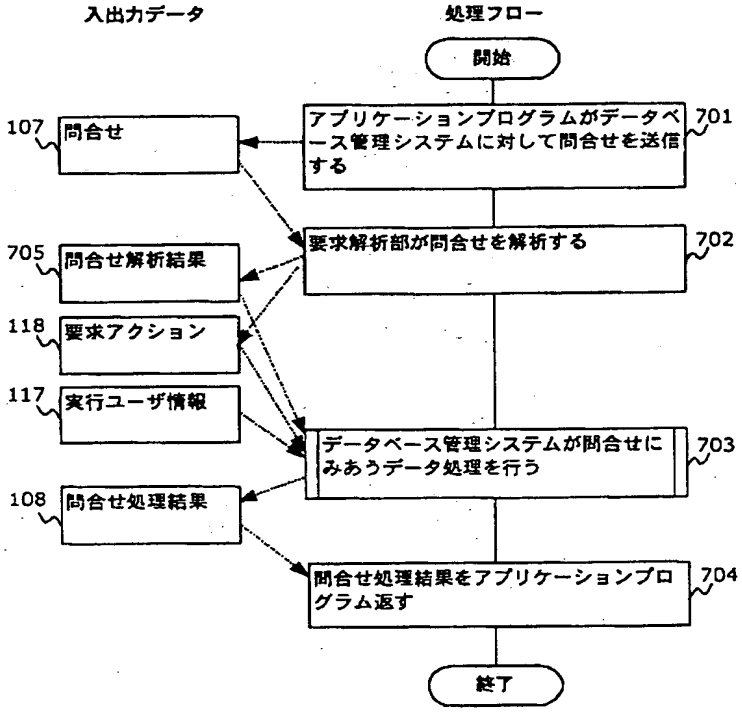
【図6】

データベース接続処理のフローチャート（図6）



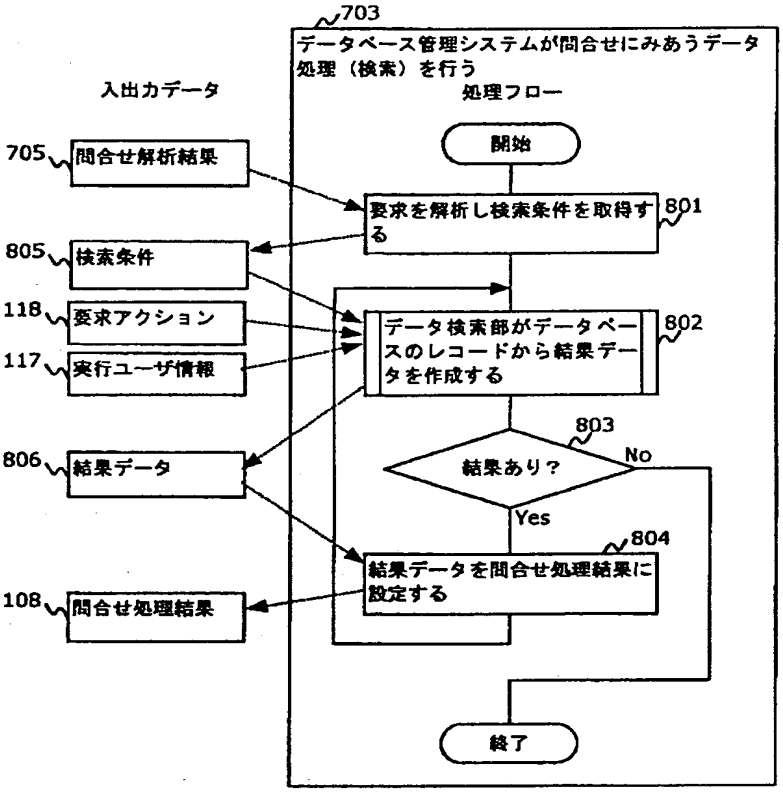
【図7】

データベース問合せ処理のフローチャート（図7）



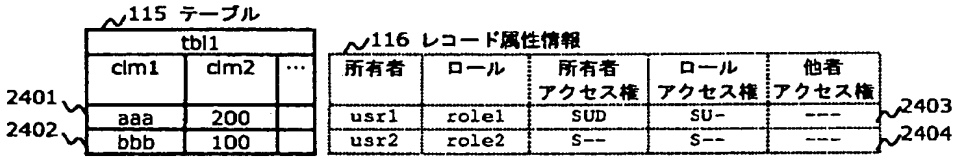
【図8】

データベース管理システムにおけるデータ検索処理のフローチャート（図8）



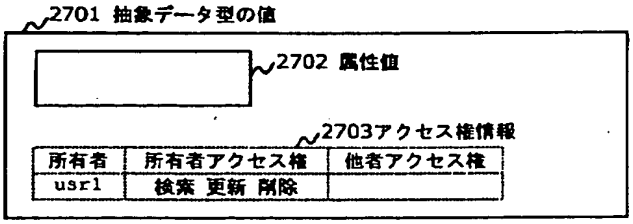
【図24】

ロールの情報を含むテーブルの構成図（図24）



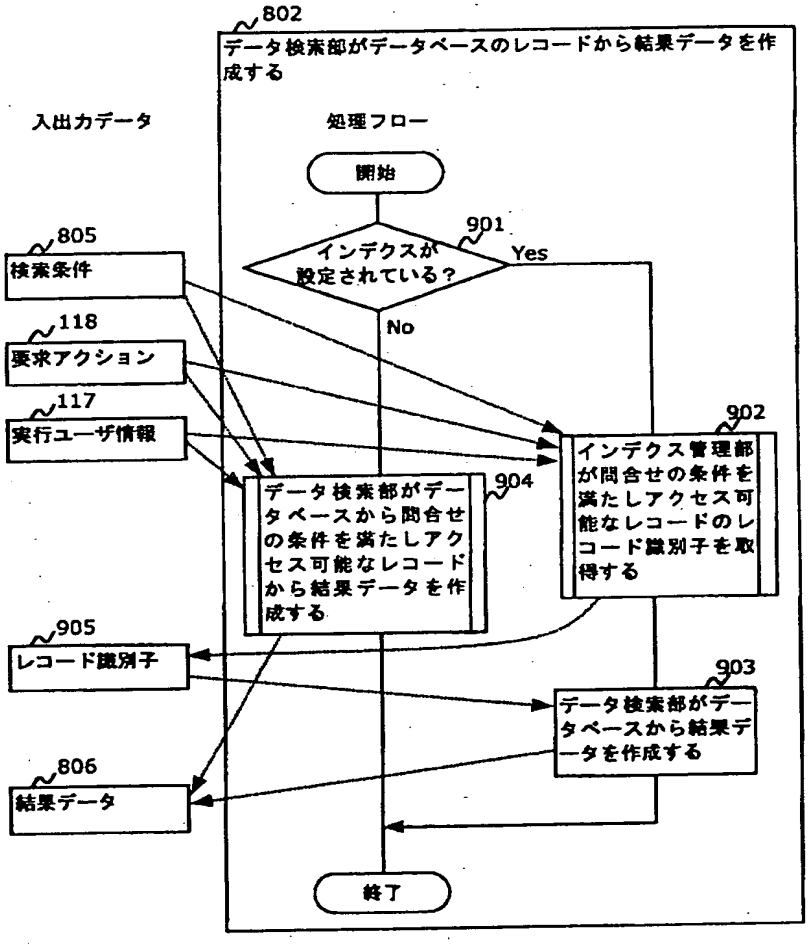
【図27】

抽象データ型のカラム値の構成図（図27）



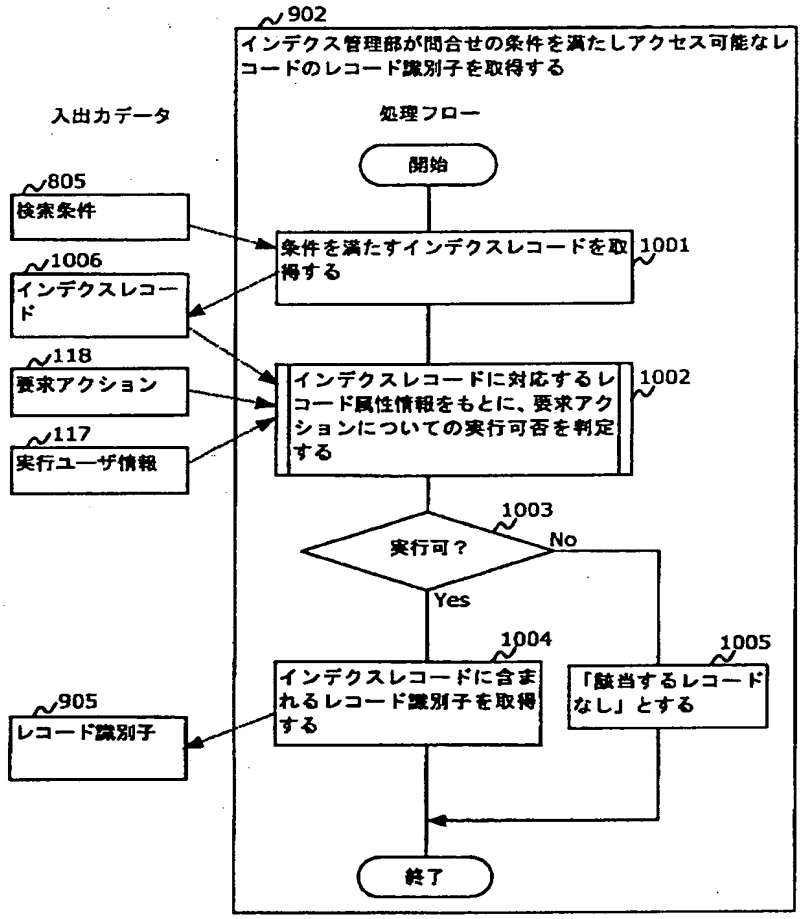
【図9】

データ検索部におけるデータ検索処理結果作成のフローチャート（図9）



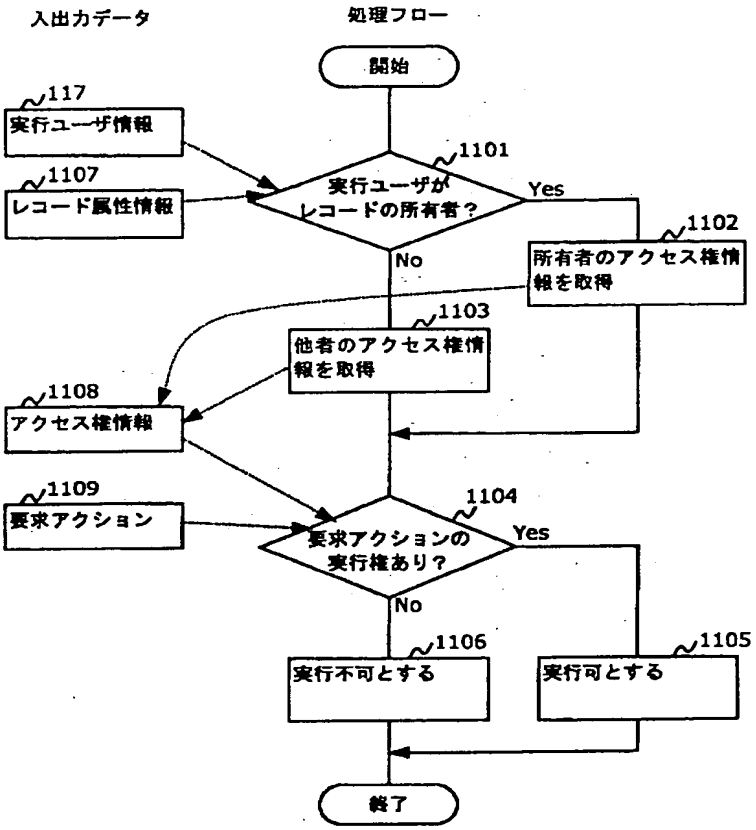
【図10】

インデクス管理部におけるデータ検索処理のフローチャート（図10）



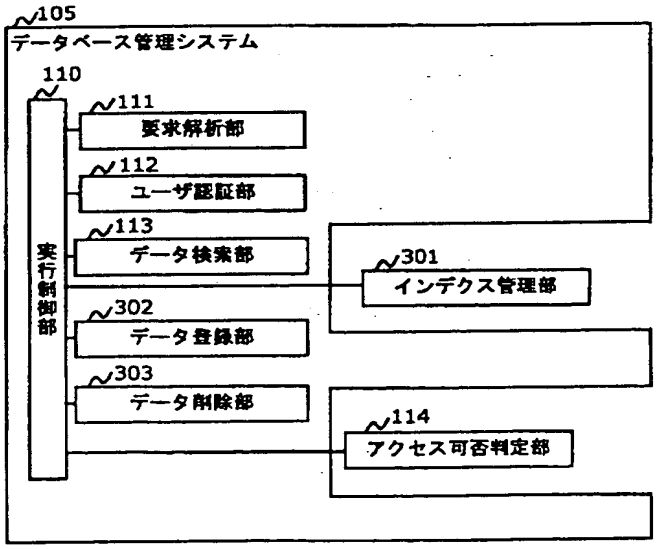
【図11】

アクセス可否判定処理のフローチャート（図11）



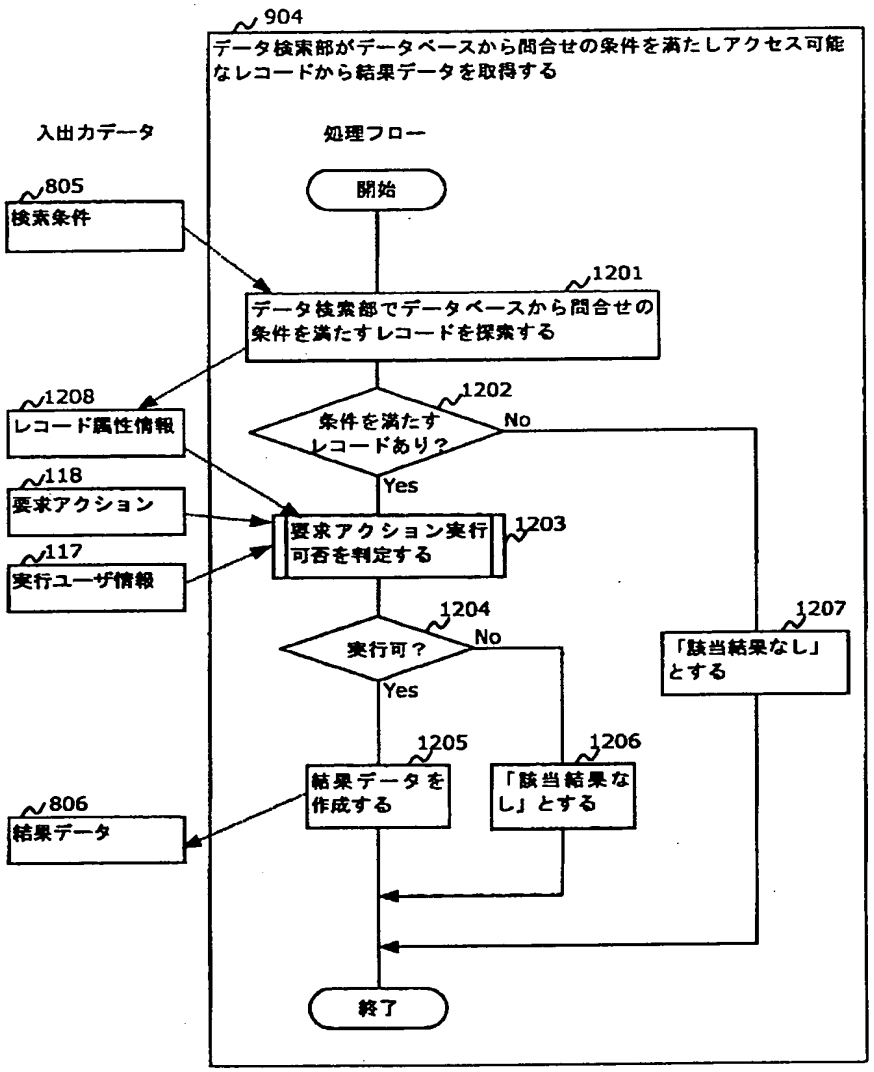
【図25】

アクセス制御を行うモジュールを独立させたデータベース管理システムの構成図（図25）



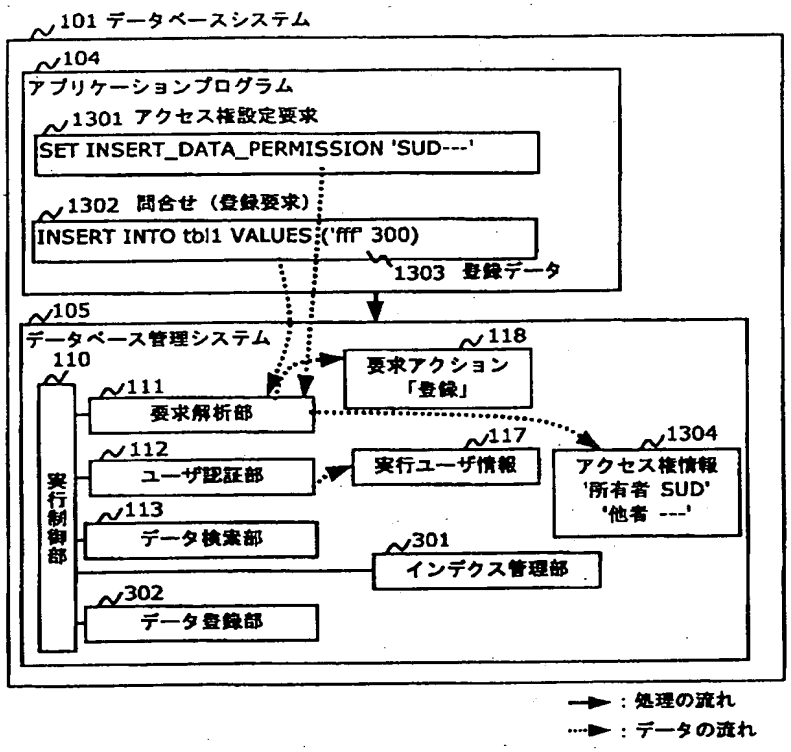
【図12】

データ検索部におけるデータ検索処理のフローチャート（図12）



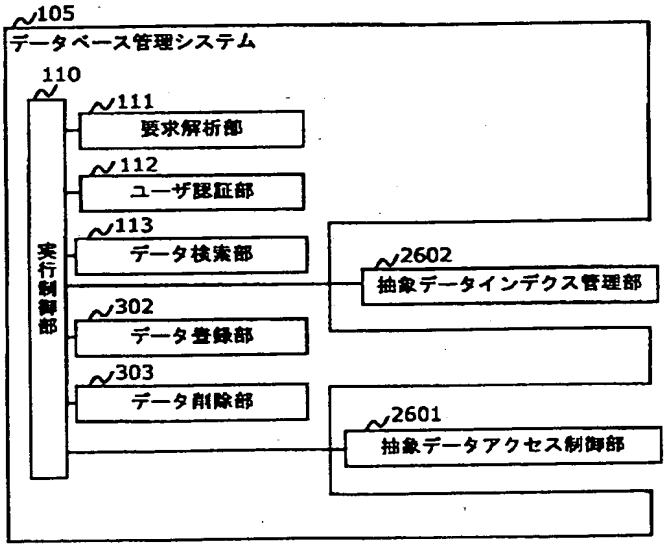
【図13】

データベースシステムの構成図（図13）



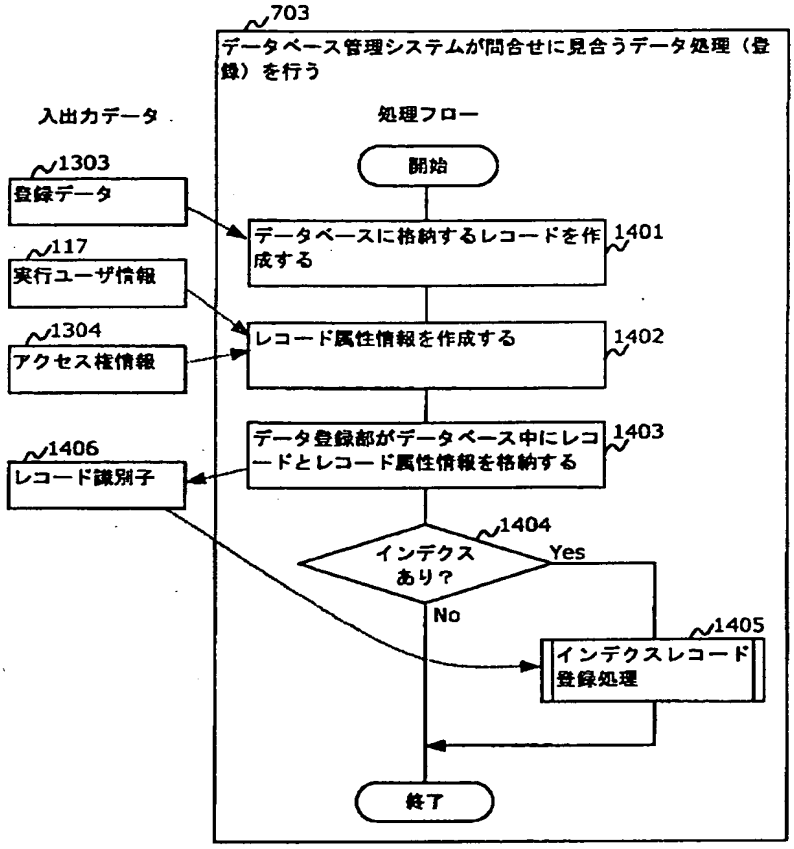
【図26】

抽象データ型のカラム値のアクセス制御を行うモジュールを用いたデータベース管理システムの構成図（図26）



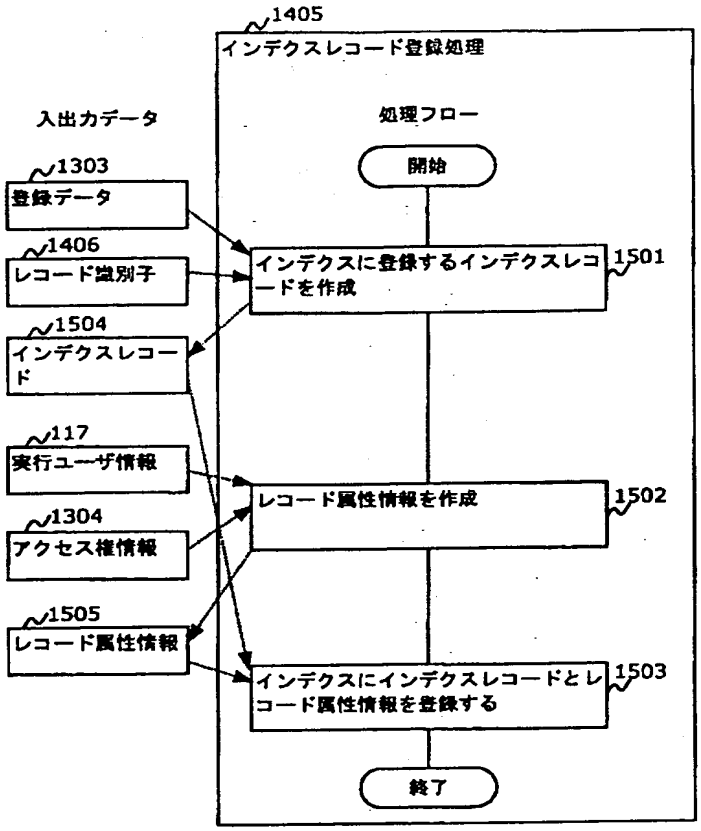
【図14】

データベース管理システムにおけるデータ登録処理のフローチャート（図14）



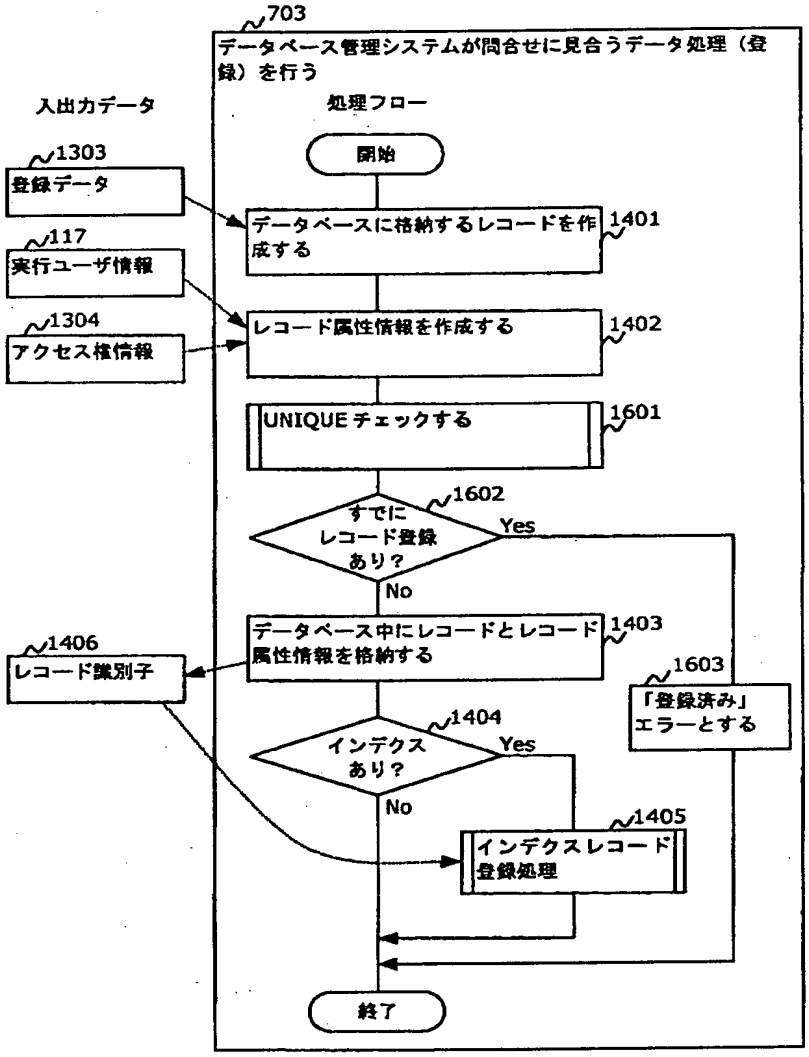
【図15】

インデクス管理部におけるインデクスレコード登録処理のフローチャート（図15）



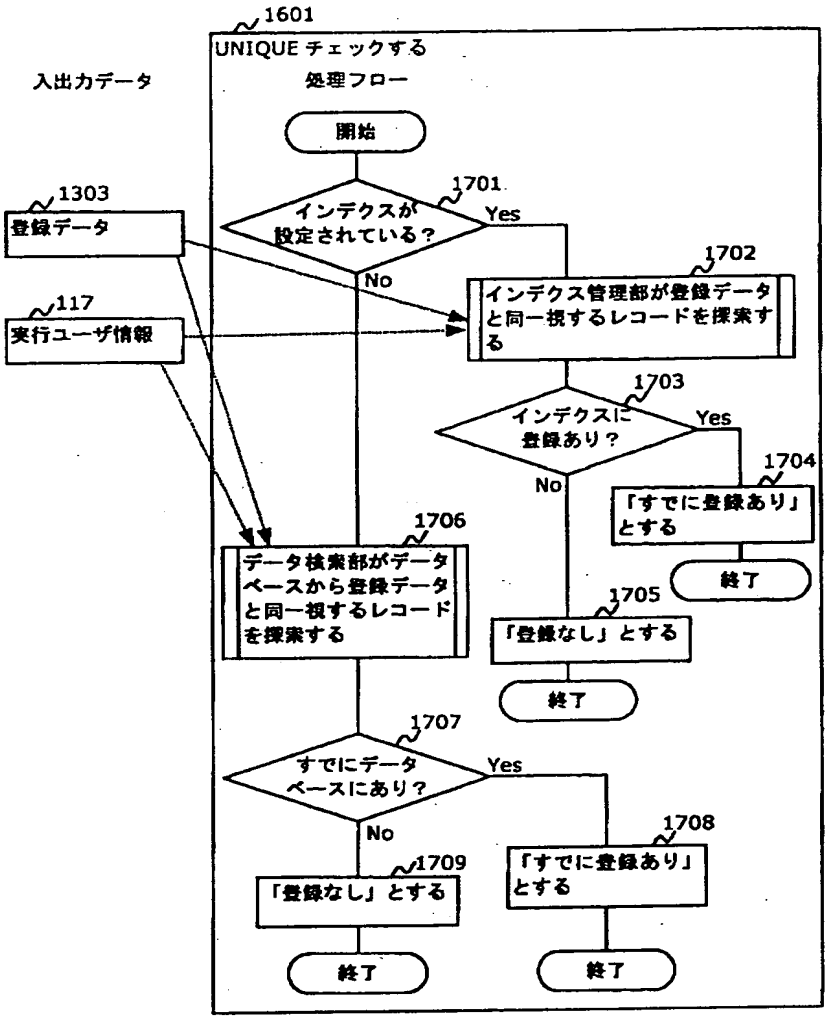
【図16】

データベース管理システムにおける UNIQUE チェックを含むデータ登録処理のフローチャート（図 16）



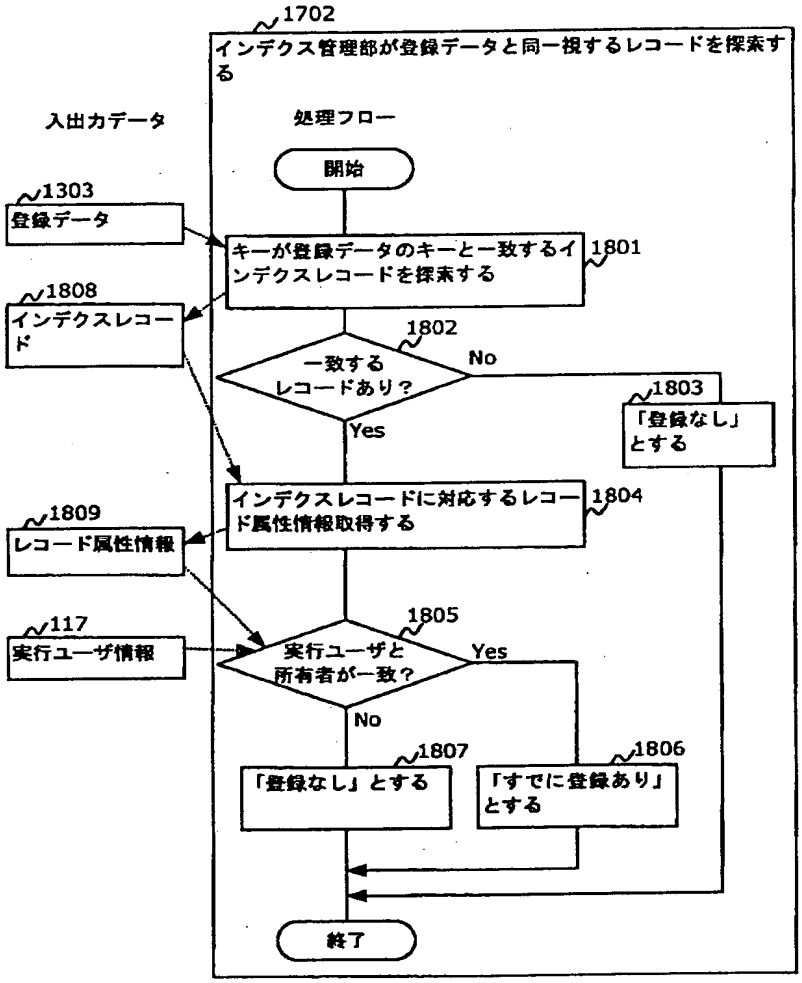
【図17】

データベース管理システムにおけるUNIQUEチェック処理のフローチャート（図17）



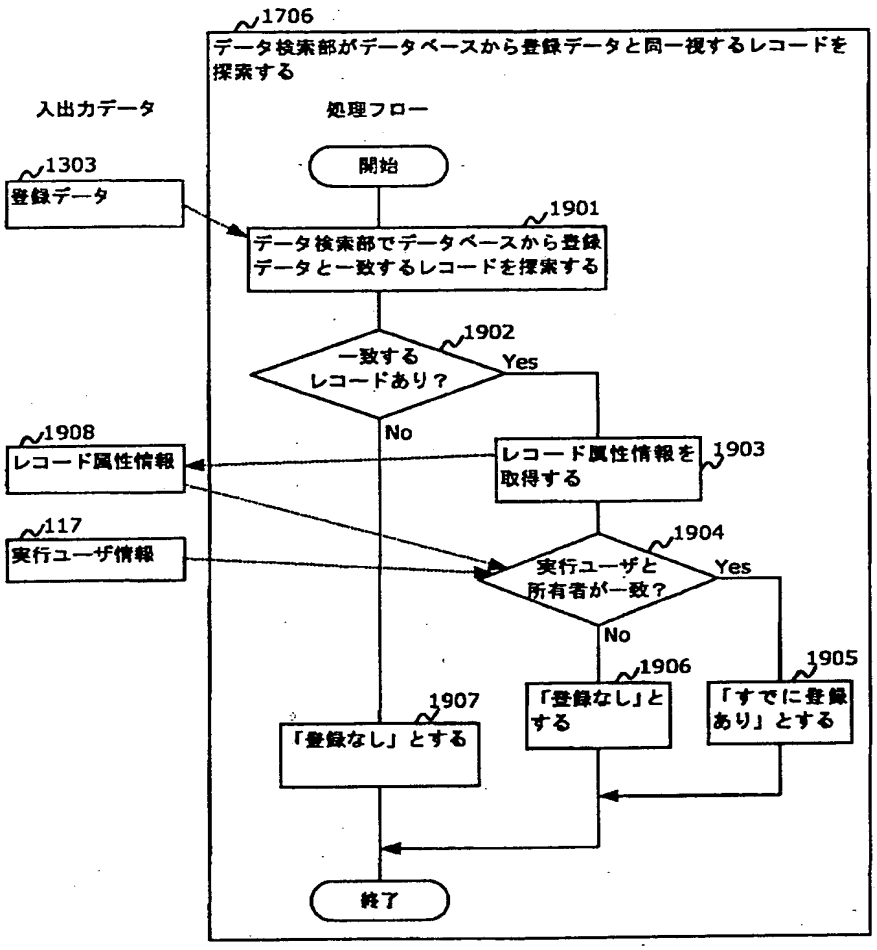
【図18】

インデクス管理部におけるUNIQUEチェック処理のフローチャート（図18）



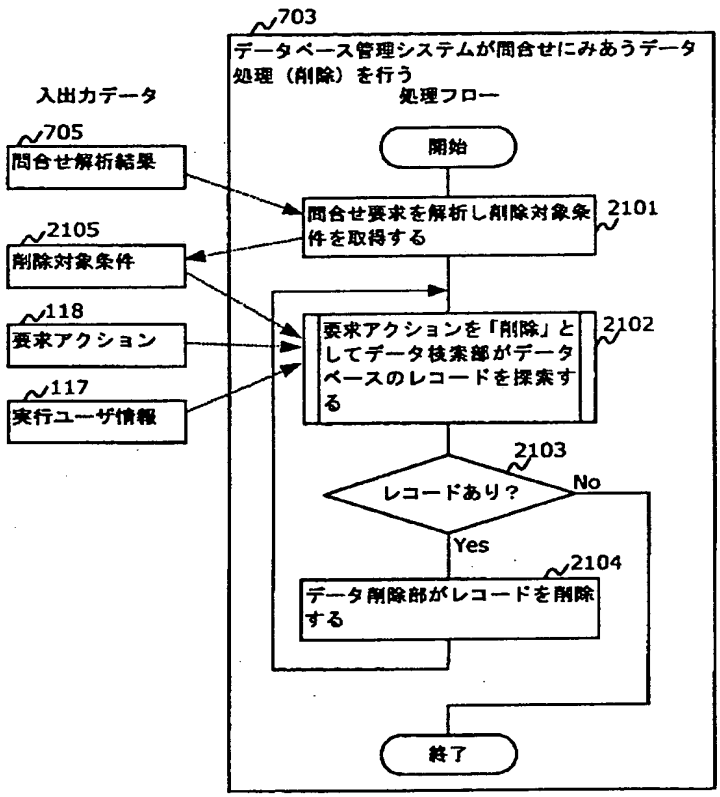
【図19】

データ検索部におけるUNIQUEチェック処理のフローチャート（図19）



【図21】

データベース管理システムにおけるデータ削除処理のフローチャート（図21）



フロントページの続き

(72)発明者　土田　正士
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所システム開発本部内

Fターム(参考)　5B017 AA01 BA06 BB06 CA16
5B082 EA07 EA11 GA11